electronics

MDPI

Article

Vulnerability Assessment and Topology Reconstruction of Task Chains in UAV Networks

Qingfeng Yue [1], Jinglei Li [1], Zijia Huang [2,*], Xiaoyu Xie [1] and Qinghai Yang [1]

[1] The School of Telecommunications Engineering, Xidian University, Xi'an 710126, China; 21181214312@stu.xidian.edu.cn (Q.Y.); jlli68@xidian.edu.cn (J.L.); xiexy96@chinaunicom.cn (X.X.); qhyang@xidian.edu.cn (Q.Y.)

[2] The 20th Research Institute of China Electronics Technology Group Corporation, Xi'an 710068, China

* Correspondence: beiyouhzj@163.com

Abstract: With the increasing complexity of environments and the diversity of task chains, individual unmanned aerial vehicles (UAVs) often struggle to satisfy the demands of task chains, including load capacity improvement, information perception, and information procession. In complex task chains involving various UAVs, such as area reconnaissance and fire rescue, any attack on critical UAVs can greatly disrupt the execution of the entire task chain by causing equipment damage or connectivity disruption. To ensure network resilience post attack, identifying vulnerable nodes in the UAV network becomes crucial. In this paper, a Vulnerability-based Topology Reconstruction Mechanism (VUTRM) is proposed to rank the importance of nodes in task chains and formulate a topology reconstruction. It consists of two parts: the first part is a Multi-metric Node Vulnerability Assessment Algorithm (MENVAL) used to rank the importance of nodes in task chains, and the second part is a Node Importance-based Topology Reconstruction Algorithm (NITRA) used to reconstruct the UAV network with the obtained node ranking. Finally, simulations carried out with simulation software demonstrate that our proposed method accurately identifies network vulnerabilities and promptly implements effective reconstruction measures to minimize network damage.

Keywords: UAV swarm network; vulnerability analysis; topology reconstruction; task chains

check for updates

Citation: Yue, Q.; Li, J.; Huang, Z.; Xie, X.; Yang, Q. Vulnerability Assessment and Topology Reconstruction of Task Chains in UAV Networks. *Electronics* **2024**, *13*, 2126. https://doi.org/10.3390/electronics13112126

Academic Editors: Weiting Zhang, Chuan Zhang and Tong Wu

Received: 24 April 2024
Revised: 20 May 2024
Accepted: 23 May 2024
Published: 29 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned aerial vehicles (UAVs) were initially predominantly used to track targets in complex environments [1]. Subsequently, their utility has expanded into civilian domains, permeating various facets of everyday life. These civilian applications encompass real-time monitoring, provision of wireless connectivity, remote sensing, search and rescue operations, cargo transportation, precision agriculture, and the surveillance of civil infrastructure locations [2]. With the growing complexity of task requirements, it has become difficulty for the independent operation of a single UAV to meet demands. To solve this problem, UAVs are employed for execution and data exchange through communication links.

The interconnectivity, communication capabilities, and interoperability among diverse types of UAVs empower them to undergo agile reconstruction as the need arises, thereby significantly bolstering the reliability and resilience of the overall system. However, due to malicious interference and complex flight scenarios, UAVs may lose connection with neighboring nodes, rendering them unable to execute tasks and causing node failures in task chains [3]. Given the limited network resources, it is crucial to find the key UAVs in the network and apply topology reconstruction. Therefore, the vulnerability analysis and topology reconstruction within task chains have constituted pivotal research challenges in recent years.

Vulnerability analysis is crucial prior to rational allocation of limited network resources. Highly important UAVs are usually assigned critical or urgent tasks, ensuring a swift and efficient response during pivotal moments [4]. This ranking facilitates optimized resource

allocation, ensuring that these UAVs receive the most dependable network resources available. Such rankings often rely on relevant parameters within the UAV network topology. Nodes with high degrees in the network typically connect to more nodes, making them pivotal points for information dissemination and task coordination [5]. In addition to the topological importance of UAVs within the communication network, we also need to consider their functional importance. Nodes play different roles in the task chain, and their importance is influenced by multiple factors, including the number of UAVs of the same type, the number of different types of connected UAVs, and the number of task chains they participate in. Some nodes may not be highly important in terms of communication topology but play crucial roles in task chain execution. However, in existing research papers, there are few studies that analyze the vulnerability of nodes based on their functional attributes.

Despite the protection measures in place for UAVs within the network, the issue of node disconnections may still arise in complex scenarios, highlighting the importance of promptly implementing repair measures for networks experiencing node failures. Topology reconstruction is one of most effective repair measures, which involves adjusting the network's connectivity structure to enhance overall robustness and adaptability. When one or more nodes fail, topology reconstruction allows the network to rapidly adapt to changes, redistributing resources and paths, thus minimizing the impact of failures and ensuring the network's stable operation and communication efficiency. In a constantly changing environment, such dynamic adjustments are crucial for the sustained effective operation of complex network systems. However, there is often a risk of excessive adjustment, leading to wastage of resources.

In this paper, we propose the Vulnerability-based Topology Reconstruction Mechanism (VUTRM), which comprises two key components: the Multi-metric Node Vulnerability Assessment Algorithm (MENVAL) for ranking node importance based on communication and functional attributes and the Node Importance-based Topology Reconstruction Algorithm (NITRA) for rebuilding the network considering node vulnerabilities. MENVAL combines metrics like node degree, betweenness centrality, and closeness centrality with functional attributes in order to assign importance scores. NITRA employs different reconstruction strategies for vulnerable and non-vulnerable nodes, optimizing network resilience and energy efficiency during failures.

The rest of the paper is organized as follows. Section 2 delves into relevant research on vulnerability node detection and reconstruction algorithms. Section 3 introduces the system model and problem formulation discussed in this paper. Section 4 details the VUTRM. Experimental results and evaluations are presented in Section 5. Finally, the entire paper is summarized in Section 6.

2. Related Work

Early research mainly focused on communication transmission between UAVs, aiming to ensure the normal operation of the network through more efficient data transmission methods. Researchers such as Mahdi Asadpour have proposed the concept of a multi-hop micro-aerial vehicle network, which focused on addressing the primary challenges of data packet forwarding in practical applications [6]. With growing concerns about secure transmission between UAVs, researchers like Bin Li introduced physical layer security into UAV organizational networks, which has become a common method to address the transmission characteristics of wireless channels [7]. Simultaneously, the identification of vulnerable nodes in UAV networks has also become a key research area, which aims to identify key UAV nodes in the network and provide them with special protection.

Typically, methods for identifying vulnerable nodes primarily originate from communication topology attributes, focusing on a node's position in the network and its connectivity with other nodes. Among these methods, degree centrality [8] is the most commonly used approach in vulnerable node identification, which primarily considers the number of a node's neighbors and has been proven effective in identifying vulnerable nodes

based on neighbor relationships [9]. Poulin et al. introduced the K-shell decomposition method, addressing the issue that a node's degree does not necessarily indicate its centrality. Additionally, identification methods based on closeness centrality [10], which measures a node's vulnerability by calculating the average shortest distance between the node and the remaining nodes in the network, have been proposed. Katz centrality, proposed in Ref. [11], is based on closeness centrality and considers other paths between nodes. Betweenness centrality takes into account the number of shortest paths passing through a node in the entire network, and a node's vulnerability is proportional to the number of paths involving that node. Ref. [12] extended betweenness centrality to directed networks, verifying its effectiveness under the same network topology. Additionally, Refs. [13,14] addressed the randomness issue in betweenness centrality by introducing a random walk-based centrality metric for identifying important nodes. In addition to discussing the communication attributes of UAVs, we also need to consider the operational characteristics that UAVs undertake during task execution. In certain task scenarios, UAVs often play diverse roles, which are typically based on the concept of the OODA (Observe, Orient, Decide, Act) loop. The method in Ref. [15] combines the concepts of betweenness and information functional chains to measure the importance of nodes and links but does not consider link integrity. On the other hand, Ref. [16] evaluates UAV network nodes and loops at the OODA loop level by considering the network diffusion (ND) and influence in the loop (NIO) of nodes. However, the algorithm's complexity is relatively high, making it challenging to meet real-time requirements in rapidly changing scenarios. Ref. [17] abstracts the elements in the operational system and their information connection relationships, constructing the network structure of the operational system. This paper assesses nodes based on aspects such as the node's substitutability, local connectivity, and operational effectiveness, providing a node ranking.

Once vulnerable nodes in the network are identified, selecting an appropriate network reconstruction strategy becomes crucial. Network vulnerability reconstruction algorithms are generally classified into two categories: large-scale and small-scale topology reconstruction algorithms. However, considering the relatively small scale in modern UAV task scenarios, large-scale recovery algorithms are less applicable. Therefore, we mainly focus on studying small-scale network topology reconstruction algorithms, especially targeting scenarios with a small number of UAVs in the network and individual node failures. Topology reconstruction algorithms, such as the Redundant Node Movement algorithm [18], the MCDS (Minimum Connected Dominating Set) algorithm [19], and the RIM (Resilient Information Matrix) algorithm [20], do not include the step of evaluating node importance. While they can reduce message transmission overhead to some extent, they also involve many unnecessary repairs. Subsequent research began to consider the importance of nodes. In the DARA (Degree and Relative Distance Aware) algorithm [21], reconstruction algorithms are selected based on the degree of neighboring nodes and the distance from the faulty nodes. However, this algorithm is prone to excessive substitution, and each node requires a substantial amount of network information. The PADRA (Partition-Aware Dominating Set-based Recovery Algorithm) [22] classifies nodes into two categories using the minimum connected dominating set, reducing total movement distance and energy consumption. However, it introduces a depth-first algorithm, leading to longer iteration cycles. The LeDiR (Least Disturbance Routing) algorithm [23] takes repair measures only after node failure, relocating the minimum number of nodes to restore the network through block movement measures. In conclusion, existing reconstruction algorithms lack discussion of node importance and fail to propose differentiated allocation schemes to proactively prevent node failures.

We propose the Vulnerability-based Topology Reconstruction Mechanism (VUTRM) to rank nodes and formulate a topology reconstruction. It consists of two parts: the first part is the Multi-metric Node Vulnerability Assessment Algorithm (MENVAL) used to rank the importance of nodes in the network, and the second part is the Node Importance-based

Topology Reconstruction Algorithm (NITRA) used to reconstruct the UAV network by the obtained node ranking. The main contribution in the VUTRM is summarized as follows:

- In MENVAL, we consider both the communication and functional topological attributes of nodes to obtain their network importance. Communication attributes encompass node degree, betweenness centrality, and closeness centrality, evaluating a node's connectivity significance. Meanwhile, functional attributes assess a node's role in task chains. By combining these attributes, MENVAL assigns an importance score to each node.
- In NITRA, we implement distinct topological reconstruction strategies for vulnerable and non-vulnerable nodes, aimed at addressing node failure issues in UAV networks. For vulnerable nodes, a relatively complex reconstruction strategy is employed to minimize their impact on the network when they fail. Conversely, for non-vulnerable nodes, a simpler strategy is used to reduce unnecessary energy consumption. This differentiation ensures effective functional recovery in case of node failures while minimizing network resource utilization.

3. System Model and Problem Formulation

Before conducting vulnerability analysis on UAV networks, it is necessary to model the system and formulate the problem. System modeling is divided into three parts: UAV nodes, communication links, and task chains. Additionally, an analysis of the primary issues related to node failures is conducted to facilitate the development of subsequent research content.

3.1. System Model

(1) UAV Nodes

In the context of UAV network nodes, we categorize the nodes according to the OODA theory into 'Observe', 'Orient', 'Decide', and 'Act'. As shown in Figure 1, four types of UAVs play different roles in the scenario, collectively forming the entire task chain. The functionalities of the four categories of nodes are as follows:

- **Observe**: These nodes are responsible for collecting data and information from the environment. They gather sensory data and observations from their surroundings.
- **Orient**: The Orient nodes process the information collected by the Observe nodes. They analyze and interpret the data to gain a better understanding of the current situation and context.
- **Decide**: Decide nodes make decisions based on the information provided by the Orient nodes. They evaluate the available options and select the most appropriate course of action.
- **Act**: Act nodes execute the decisions made by the Decide nodes. They carry out the chosen actions, which may involve controlling the movement or operation of unmanned aerial vehicles or other tasks.

We use V^S, V^I, V^D, and V^J to respectively represent the sets of Observe, Orient, Decide, and Act nodes, with V representing all the unmanned aerial vehicle nodes in the network: $V = V^S \bigcup V^I \bigcup V^D \bigcup V^J$.

(2) Communication Links

In the context of a UAV network, edges in the network represent the communication links between UAV nodes. During UAVs' flight, the communication range between UAV nodes is calculated, and it is defined that two UAVs can only engage in wireless communication if they are both within each other's communication range. After modeling the nodes, UAVs are deployed. Subsequently, the UAVs undergo a specified number of motion iterations. In each iteration, the generated network topology is recorded. The connection status $E_{ij}^{n_t}$ between nodes i and j at the n_t-th iteration is defined as follows:

$$E_{ij}^{n_t} = \begin{cases} 1, & \text{when there is a connection and i < j} \\ 0, & \text{otherwise.} \end{cases}$$

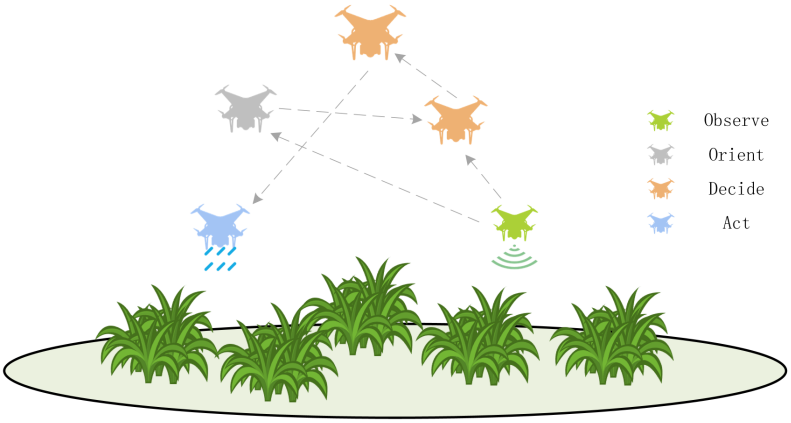


Figure 1. Scenario for a UAV task chain.

(3) Task Chains

In the OODA, task chains consist of four types of UAVs. If we study the functional connections between each set of two UAVs, the types of edges could amount to 16. However, if we study every possible connection among UAVs, the number of connection types involved will be extremely vast, leading to a significant workload. To focus more on the research emphasis of this paper, we primarily studied five types of directional connections within the UAV network: Perception–Collection Edge (T^S), Perception–Control Edge (T^S), Collection–Control Edge (T^I), Command–Control Edge (T^D), and Control–Execution Edge (T^J). The Perception–Collection Edge involves communication links from Observe nodes to Orient nodes, where Observe UAVs transmit collected information to Orient nodes ($T^S : V^S \rightarrow V^I$). The Perception–Control Edge represents direct communication links from Observe UAVs to Decide UAVs for decision-making and assessment, without going through Orient nodes ($T^S : V^S \rightarrow V^D$). The Collection–Control Edge signifies communication links from Orient nodes to Decide nodes, where Orient nodes forward gathered information from Observe nodes after computation and processing ($T^I : V^I \rightarrow V^D$). The Command–Control Edge represents communication links between Decide nodes with a hierarchical relationship among them ($T^D : V^D \rightarrow V^D$). Lastly, the Control–Execution Edge denotes communication links from Decide nodes to Act nodes, where Decide nodes transmit decisions to Act nodes, and Act nodes arrange tasks accordingly based on received information ($T^J : V^D \rightarrow V^J$).

These five types of communication links can be combined to form a complete task chain, and the combination method is illustrated in Figure 2. It can be observed that the composed task chains mainly consist of four types: $V^S - V^I - V^D - V^J$, $V^S - V^D - V^D - V^J$, $V^S - V^D - V^J$, and $V^S - V^I - V^D - V^D - V^J$. Their corresponding links are represented as $T^S - T^I - T^J$, $T^S - T^J$, $T^S - T^D - T^J$, and $T^S - T^I - T^D - T^J$.

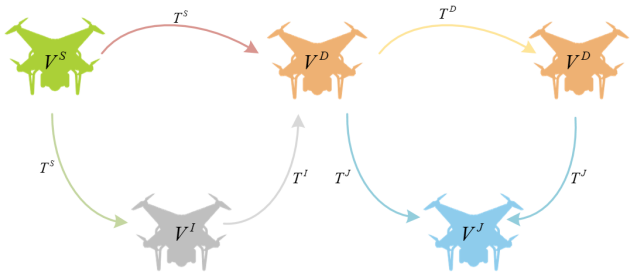


Figure 2. An example of communication links.

(4) Network Topology

The network topology matrix, denoted as E^{n_t}, is used to represent the connection relationships in the UAV network, specifying that two UAVs can only transmit information when they are within each other's communication range. Table 1 illustrates the communication connections in the network shown in Figure 3 using this matrix. The figure depicts a network composed of five UAVs, including one Observe node, one Orient node, two Decide nodes, and one Act node. Information transmission between UAVs is only possible when they are within each other's communication range. For example, Node 1 has communication connections with Nodes 2 and 3, so the values in the first row of the table are 1 for columns 2 and 3 and 0 for other columns. The situation is similar for other nodes. The topology matrix effectively reflects the communication topology in the network.

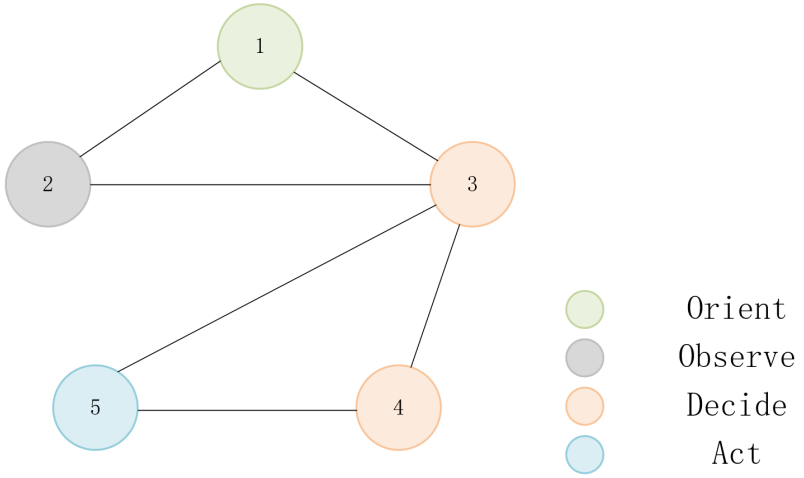


Figure 3. A software-generated network topology diagram.

Table 1. The upper-triangular conditional adjacency matrix E^{n_t}.

Node	1	2	3	4	5
1	0	1	1	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	0	0	0	0	1
5	0	0	0	0	0

3.2. Problem Formulation

Task chains play a crucial role in distributed systems, consisting of a series of interdependent tasks or operations aimed at achieving specific objectives. In such systems, topology connectivity is paramount, as it ensures communication and collaboration among the nodes within the task chain, ensuring smooth execution of tasks. However, task chains are vulnerable to various attacks or disruptions, such as network attacks, node failures, etc., which may lead to interruptions or abnormal execution of tasks. Therefore, conducting vulnerability analysis and topology reconstruction is essential for identifying and addressing potential issues, thus enhancing the reliability and stability of task chains. Through these measures, the resilience and robustness of task chains can be effectively strengthened, ensuring their smooth operation in complex network environments.

The main goal of this study is to explore how to ensure maximum remaining task integrity after node failures in UAV scenarios. To address this issue, we divide it into two steps: assessing vulnerable nodes and reconstructing network topology. The former step is aimed at assessing the importance of the failed nodes within the network, meaning they play crucial roles in the network and their failures could have a significant impact on the entire network. The latter discusses how to reconstruct connections and communication

paths in the network when node failures occur in order to ensure that the network can still effectively carry out its tasks.

In the first step, we aim to identify important nodes within the network. To test whether the nodes identified by the algorithm are vulnerable, we will remove them and calculate the remaining task integrity of the network after their removal. The remaining task connectivity R is defined as the ratio of the number of task chains in the network after node failures to the initial number of task chains, which is expressed as follows:

$$R = \frac{1}{l_0} \sum_{i=1}^{N} l_i^{n_t}, \tag{1}$$

where $l_i^{n_t}$ represents the number of task chains passing through node i at the n_t-th iteration in the network after each topological update, and l_0 represents the number of all task chains in the network before the node attack. Generally, the larger the importance of a node, the greater the impact on network task execution when the node fails, resulting in lower remaining task chain integrity.

In the second step, we also use the network remaining task chain integrity to represent the repair effectiveness of the algorithm and use the movement distance to indicate how far candidate UAVs have to move for topological reconstruction. The greater the distance a UAV flies, the more energy it consumes, and conversely, the shorter the distance, the less energy is consumed. Therefore, the objective of the reconstruction algorithm is to minimize the energy consumption of the swarm while maximizing the network's remaining task integrity after reconstruction.

4. Vulnerability-Based Topology Reconstruction Mechanism

As described above, the VUTRM algorithm consists of two parts: MENVAL evaluates the importance of nodes by assessing their communication topology attributes and functional topology attributes, while NITRA maintains the stability of the UAV network by employing different topology reconstruction strategies for vulnerable and non-vulnerable nodes. This section provides a detailed explanation of these two algorithms.

4.1. Multi-Metric Node Vulnerability Assessment Algorithm

MENVAL evaluates the importance of nodes based on both their communication topology attributes and functional topology attributes. The former focuses on the geographical significance of nodes within the network, while the latter emphasizes their importance within the entire task chain.

4.1.1. Importance of Communication Topology

The importance of communication topology focuses on the significance of a node's topological position within the network and the extent of its connectivity with other nodes, without considering the node's functionality. The topological importance of nodes in the network consists of three components: node degree, node betweenness, and node closeness.

(1) Node Degree

Node degree is the most fundamental metric for identifying important nodes: the more neighbors a node has, the higher its degree centrality is, indicating its greater importance. In a UAV's network, it is only necessary to count the number of nodes connected to a particular node in the directed adjacency matrix E^{n_t}. The node degree $K_i^{n_t}$ of the node i at motion iteration n_t is represented as follows:

$$K_i^{n_t} = \sum_{i<j} E_{ij}^{n_t} + \sum_{i>j} E_{ji}^{n_t}. \tag{2}$$

After normalizing $K_i^{n_t}$, we obtain

$$K'^{n_t}_i = \frac{K^{n_t}_i}{K^{n_t}_{\max}}, \tag{3}$$

where $K'^{n_t}_i$ represents the normalized node degree, and $K^{n_t}_{\max}$ represents the maximum node degree among all nodes at iteration count n_t.

(2) Node Betweenness

Node betweenness is a key metric in network analysis that quantifies the importance of a node by measuring the number of shortest paths passing through it. The higher the betweenness centrality of a node, the more influential it is in controlling the flow of information or resources within the network. As shown in Figure 4, it represents a UAV network with a total of 13 nodes. According to the node degree formula, we can calculate the degree of each node, where $K'^{n}_2 = K'^{n}_3 = K'^{n}_4 = \frac{1}{6} > K'^{n}_1 = \frac{1}{8}$. From the perspective of node degree alone, Nodes 2, 3, and 4 are all more important than Node 1.

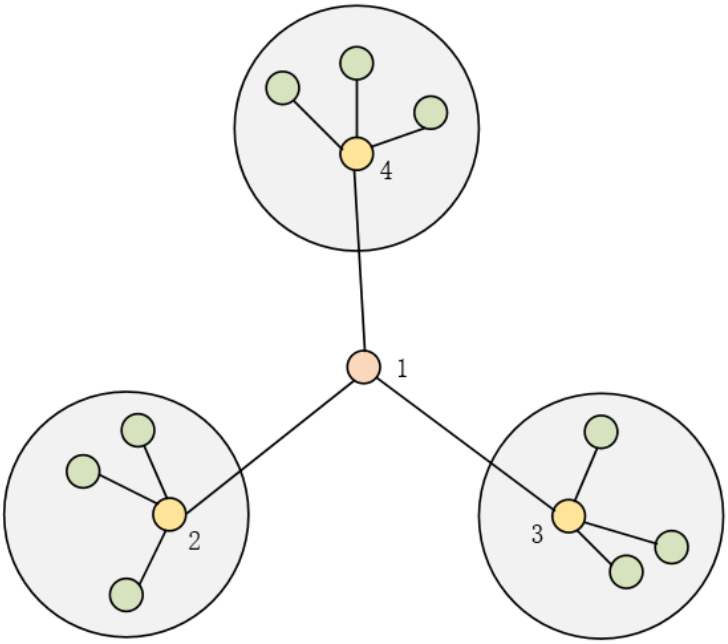


Figure 4. Illustration of node betweenness.

However, it is evident that Node 1 is the hub of the entire network. In practical networks such as UAV networks or social networks, hub nodes are often the most crucial. Therefore, we calculate the number of shortest paths in which each node participates to analyze the importance of nodes in the entire network. The node betweenness $B^{n_t}_i$ of the node i at iteration count n_t is represented as follows:

$$B^{n_t}_i = \frac{g^{n_t}_{ist}}{g^{n_t}_{st}}, \tag{4}$$

where $g^{n_t}_{st}$ represents the number of shortest paths from node s to node t at the n_t-th iteration, and $g^{n_t}_{ist}$ represents the number of shortest paths passing through node i. After normalizing $B^{n_t}_i$, we obtain the following:

$$B'^{n_t}_i = \frac{B^{n}_i}{B^{n_t}_{\max}}, \tag{5}$$

where $B'^{n_t}_i$ represents the normalized node betweenness, and $B^{n_t}_{\max}$ represents the maximum node betweenness among all nodes at iteration count n_t.

(3) Node Closeness

Closeness centrality is a measure in network analysis that quantifies how quickly a node can interact with other nodes in the network. It is calculated based on the average shortest path length from a node to all other nodes in the network. Nodes with higher closeness centrality are considered more central and have shorter average distances to other nodes, indicating their greater accessibility and influence on the network. Assuming $d_{ij}^{n_t}$ represents the number of hops from node i to another node j at iteration n_t in a topology network, and the average distance from this node to all remaining nodes in the network is denoted as $\mathrm{d}_i^{n_t}$, it can be represented as follows:

$$d_i^{n_t} = \frac{1}{N} \sum_{j=1}^{N} d_{ij}^{n_t}. \tag{6}$$

It is obvious that when d is smaller, the average distance from node i to all other nodes in the network is closer. To some extent, it indicates that the topology of node i is of greater importance. Therefore, the reciprocal of d, denoted as c, is defined as the closeness centrality of node i in the network, and the formula is expressed as follows:

$$C_i^{n_t} = \frac{1}{d_i^{n_t}} = \frac{N}{\sum_{j=1}^{N} d_{ij}^{n_t}} \tag{7}$$

$$C'^{n_t}_i = \frac{C_i^{n_t}}{C_{\max}^{n_t}}, \tag{8}$$

where $C'^{n_t}_i$ represents the normalized node closeness, and $C_{\max}^{n_t}$ represents the maximum node closeness among all nodes at iteration count n_t.

According to the assigned weights for the three evaluation metrics α_K, α_B, and α_C based on task execution, the normalized topological importance $\beta_{\mathrm{i}}^{n_t}$ of node i at iteration n_t can be represented as follows:

$$\beta_{\mathrm{i}}^{n_t} = \alpha_K K'^{n_t}_{\mathrm{i}} + \alpha_B B'^{n_t}_i + \alpha_C C'^{n_t}_i, \tag{9}$$

where three weight coefficients satisfy the following conditions:

$$\alpha_K + \alpha_B + \alpha_C = 1. \tag{10}$$

4.1.2. Importance of Functional Topology

For the evaluation of node importance in task chains, it is necessary to consider the functions of UAVs. In task chains systems, not all paths are meaningful; only those that form the OODA task chain can reflect the efficacy in task chains. Therefore, the importance of functional topology comprehensively considers three key aspects of elements, including substitutability, node connectivity, and efficiency importance of nodes. Through these measurements, we are able to provide a comprehensive assessment of the importance of elements in the network, offering valuable guidance for optimizing and enhancing the task chains.

(1) Importance of Substitutability

The substitutability of a node indicates whether other UAVs in the network can replace its functionality in case of failure or attack. The substitutability is related to the number of nodes in the network: the more nodes of the same type, the higher the substitutability; conversely, the fewer nodes of the same type, the greater the system's dependency on that UAV. Therefore, substitutability can be represented as follows:

$$\gamma_i^{n_t} = |V_X|, \quad i \in X \text{ and } X \in \{S, I, D, J\}, \tag{11}$$

where V_X represents the set element of X-type UAVs when the iteration count is N; $|\cdot|$ denotes the quantity of UAVs in the specified set; and S, I, D, and J represent the sets of

UAVs for Observe, Orient, Decide, and Act, respectively. The normalized node's importance of substitutability can be represented as follows:

$$\gamma'^{n_t}_i = \frac{|V^{n_t}_X|}{N}, \quad i \in X \text{ and } X \in \{S, I, D, J\}. \tag{12}$$

(2) Importance of Connectivity

Importance of connectivity is used to describe the degree of connectivity between a node and other types of nodes in the UAV network. The higher the degree value of a node, the greater its level of interaction with other nodes. Taking an example, the connectivity importance $d^{n_t}_i$ of observation UAV node i at the n_t-th network iteration is defined as the number of nodes in three categories: Orient, Decide, and Act, that are connected to node i. Since the matrix is an upper triangular matrix, it can be expressed as follows:

$$d^{n_t}_i = \sum_{j\in I} E^{n_t}_{ij} + \sum_{j\in I} E^{n_t}_{ji} + \sum_{j\in D} E^{n_t}_{ij} + \sum_{j\in D} E^{n_t}_{ji} + \sum_{j\in J} E^{n_t}_{ij} + \sum_{j\in J} E^{n_t}_{ji}, i \in S. \tag{13}$$

The normalized node's importance of connectivity can be represented as follows:

$$d'^{n_t}_i = \frac{d^{n_t}_i}{d^{n_t}_{\max}}, \tag{14}$$

where $d'^{n_t}_i$ represents the normalized importance of connectivity, and $d^{n_t}_{\max}$ represents the maximum importance of connectivity among all nodes at iteration count n_t.

(3) Importance of Efficiency

The more task chains a node participates in, the more significant its impact on the overall network's task implementation effectiveness. Therefore, we analyze the number of task links each type of node participates in. For links of the Observe node and Act node, because each task chain always includes both of them, the task chains involving both can be discussed together and represented by the following formula:

$$\mathrm{b}^{n_t}_{ij} = T^S T^I T^J + T^S T^J + T^S T^D T^J + T^S T^I T^D T^J, \tag{15}$$

where $\mathrm{b}^{n_t}_{ij}$ indicates that there are $\mathrm{b}^{n_t}_{ij}$ task chains from Observe node i to Act node j at the n_t-th iteration as follows:

$$l^{n_t}_i = \sum_{j=|V^S|+|V^I|+|V^D|+1}^{N} b^{n_t}_{ij}, i \in \{1, ..., |V^S|\}. \tag{16}$$

The total number of task chains passing through the task execution node j is given by:

$$l^n_i = \sum_{j=1}^{|V^S|} b^{n_t}_{ji}, i \in \{|V^S| + |V^I| + |V^D| + 1, |V^S| + |V^I| + |V^D| + 2, ..., N\}. \tag{17}$$

For links passing through Orient nodes, they mainly involve two types of edges, which consist of $T^I T^J$ and $T^I T^D T^J$, and they can be expressed as follows:

$$c^{n_t}_{ij} = T^I T^J + T^I T^D T^J. \tag{18}$$

The quantity $c^{n_t}_{ij}$ represents the number of links from the Orient node i to the Act node j at the n_t-th iteration: $i \in \{|V^S| + 1, |V^S| + 2, ..., |V^S| + |V^I|\}$, $j \in \{|V^S| + |V^I| + |V^D| + 1, |V^S| + |V^I| + |V^D| + 2, ..., N\}$. The number of links passing through the Orient node is:

$$l_i^{n_t} = \sum_{j=|V^S|+|V^I|+|V^D|+1}^{N} c_{ij}^n \times \sum_{j=1}^{|V^S|} E_{ji}^n, \quad i \in \{|V^S|+1, |V^S|+1, ..., |V^S|+|V^I|\}. \tag{19}$$

For links of the Decide node, they mainly involve three types of edges, which are T^S, $T^S T^I$, and $T^S T^I T^D$. The number of links passing through Decide nodes can be expressed as follows:

$$d_{ij}^{n_t} = T^S + T^S T^I + T^S T^I T^D, \tag{20}$$

where d_{ij} represents the number of links from the Decide node i to the Act node j at the n_t-th iteration: $i \in \{1, 2, ..., |V^S|\}, j \in \{|V^S|+|V^S|+1, |V^S|+|V^S|+2, ..., |V^S|+|V^S|+|V^S|\}$. The number of links passing through the Orient node is:

$$l_i^{n_t} = \sum_{j=|V^S|+|V^I|+|V^D|+1}^{N} c_{ij}^{n_t} \times \sum_{j=1}^{|V^S|} E_{ji}^{n_t}, \tag{21}$$

$$i \in \{|V^S|+|V^I|+1, |V^S|+|V^I|+2, ..., |V^S|+|V^I|+|V^D|\}$$

After normalizing the node's importance for functional topology, it can be represented as follows:

$$l'^{n_t}_i = \frac{l_i^{n_t}}{l_{\max}^{n_t}}, \tag{22}$$

where $l'^{n_t}_i$ represents the normalized importance of efficiency, and $l_{\max}^{n_t}$ represents the maximum importance of efficiency among all nodes at iteration count n_t.

Based on the performed task, we allocate weights to three evaluation metrics, α_γ, α_d, and α_l. From iteration $n_t - 1$ to iteration n_t, the functional importance $W_i^{n_t}$ of node i can be represented as follows:

$$W_i^{n_t} = \alpha_\gamma \gamma'^{n_t}_i + \alpha_d d'^{n_t}_i + \alpha_l l'^{n_t}_i, \tag{23}$$

where three weight coefficients satisfy the following conditions:

$$\alpha_\gamma + \alpha_d + \alpha_l = 1. \tag{24}$$

4.1.3. Comprehensive Importance

Considering the importance of communication topology and functional topology of each node, the comprehensive topological importance ρ_i of node i at iteration n is obtained as follows:

$$\rho_i^{n_t} = \varphi_1 \beta_i^{n_t} + \varphi_2 W_i^{n_t}, \tag{25}$$

where φ_1 represents the weight coefficient for communication topology importance, and φ_2 represents the weight coefficient for functional topology importance; the sum of these two coefficients satisfies the following:

$$\varphi_1 + \varphi_2 = 1. \tag{26}$$

By calculating the arithmetic average of the accumulated comprehensive importance form the previous equations, we obtain:

$$\rho'^{n_t}_i = \frac{1}{n_t} \sum_{n=1}^{n_t} \rho_i^n. \tag{27}$$

4.2. Node Importance-Based Topology Reconstruction Algorithm

In Section 2 of this paper, we discussed the significance of studying network reconstruction algorithms that discern node importance. Additionally, we provided an in-depth background of the specific algorithm examined, outlining its theoretical basis and relevance in network analysis. Before introducing the algorithm, we made some prerequisites in our scenario that each node in the network has a unique ID obtained through technologies like GPS, ensuring that the failure of a particular node does not affect the unique IDs of other nodes. Furthermore, each node stores information about nodes within a two-hop range, including details such as distance, type, importance, and other relevant node information. Additionally, nodes possess the ability to move freely. In the event of a UAV node becoming inactive due to an attack, the network can be repaired by relocating other nodes. However, this process is constrained by the limited energy of each node, with activities such as data transmission and network maintenance consuming energy. Notably, the energy consumption for moving to reconstruct the network is much higher than for the first two activities, leading to the adoption of moving reconstruction only in essential situations.

In current research on UAV topology reconstruction, the MCMA (Minimum Cascade Mobile Algorithm) is commonly employed [24]. MCMA consists of two steps: cut-point detection and node movement reconstruction. When a node in the network fails, the algorithm performs cut-point detection.

This algorithm combines cut-point detection with node movement reconstruction, to some extent addressing the problems of excessive substitution and excessive movement distance . Based on the algorithm, we have implemented the following modifications. UAVs are categorized into distinct types, allowing only UAVs of the same type to substitute for one another. In the mobile reconstruction strategy, we prioritize the importance of failed nodes and candidate replacement nodes to ensure post-reconstruction network robustness surpasses the pre-reconstruction state, avoiding unnecessary reconstruction and repair for every failed node. While MCMA primarily focuses on reconstructing and repairing cut-points, we have recognized the potential benefit of having UAVs move shorter distances to enhance network robustness. Thus, we have designated different topological reconstruction strategies for non-vulnerable and vulnerable nodes, with the former requiring fewer system resources for scheduling compared to the latter.

Inspired by the modifications made to MCMA as described above, we propose NITRA. The process of NITRA is shown in Figure 5, and its pseudocode is shown in Algorithm 1. In this algorithm, we define the set of nodes with the same type and smaller comprehensive importance within the selected range of the failed node as the candidate node set. Before each iteration, we check whether there are nodes that have failed in the network. If there is a failed node and the node has its candidate node, we perform a movement reconstruction, which is the replacement of the least important node in the set of candidate nodes for the failed node. If there are no nodes meeting the criteria within the one-hop range of the failed node, we then determine whether to expand to the two-hop range based on whether the failed node is a vulnerable node. If the failed node is a vulnerable node, we search for candidate nodes within the two-hop range. If there are nodes that meet the criteria, we perform a movement reconstruction, and this process continues until the maximum number of iterations is reached. The time complexity of this algorithm depends on the number of edges in the network and the maximum degree of nodes, which is $O(|E| \cdot d_{\max})$. In the algorithm, iterations are performed for each failed link and for each neighboring node of each node. For each node, the algorithm requires retrieving its set of neighboring nodes, followed by calculating additional data allocation and updating data quantity based on a specific formula. Overall, the complexity of this algorithm increases with the scale of the network but is also constrained by the maximum degree of nodes.

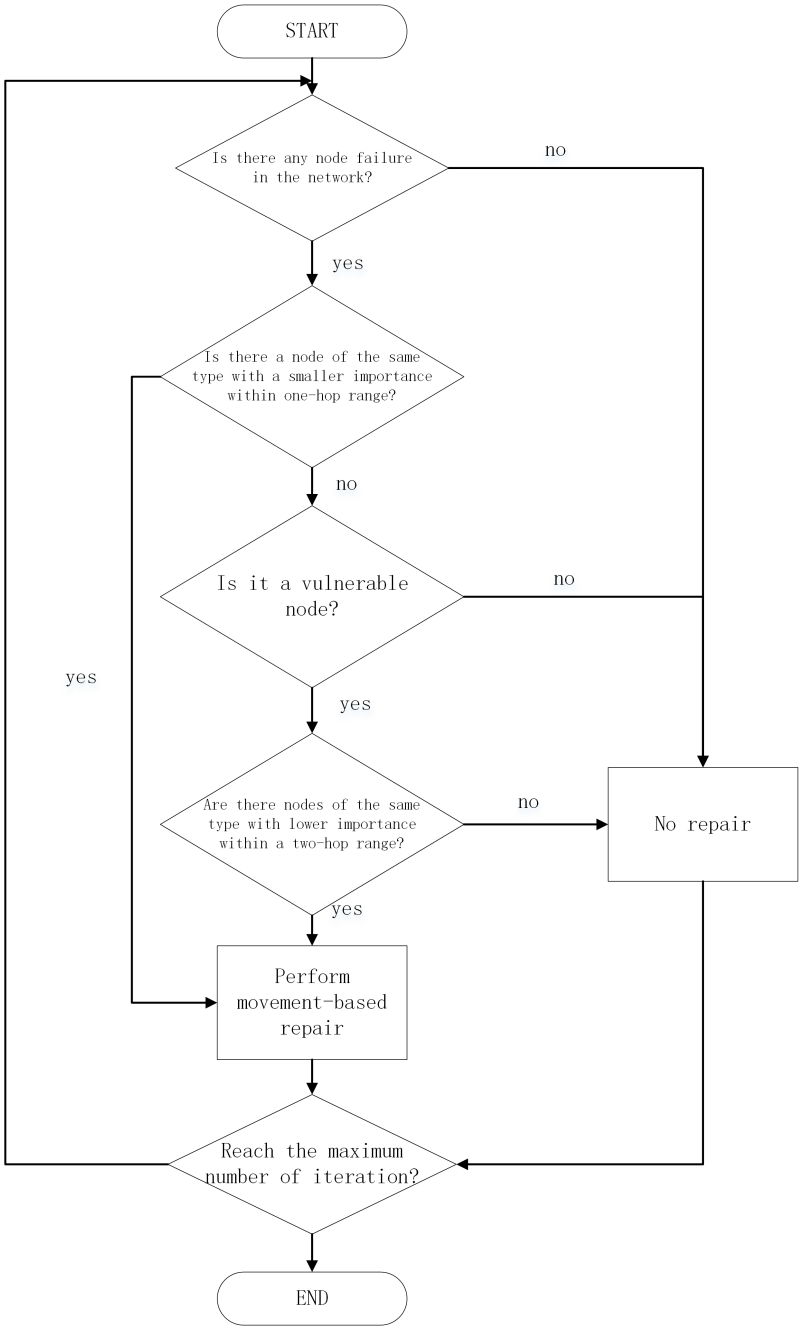


Figure 5. Process of NITRA.

As an example, the topology structure of the UAV network generated in a single simulation is shown in Figure 6. Assuming Node 23 fails, and since it is a non-vulnerable node, we only search for candidate nodes within the one-hop range. From the graph, we can see that, within the one-hop range of Node 23, Nodes 22 and 24 are present, but they are not of the same type as Node 23. Therefore, no reconstruction operation is performed for the failure of Node 23. If Node 15 fails, within its one-hop range, we find Nodes 14, 8, and 16. Since these three nodes are not of the same type as Node 15, we expand the range to two hops. Within the two-hop range of Node 15, we find Nodes 13, 9, 20, 9, 4, 7, 17, 21, and 22. Among them, Nodes 4 and 17 are of the same type, and since the importance of Node 4 is relatively lower than that of Node 17, Node 4 becomes a candidate node to replace Node 15. Node 4 is then moved to the position of Node 15. Although in this iteration, Node 2 loses Node 4 and becomes an isolated node, the dynamic nature of the network allows Node 2 to join the network in the next iteration. If no candidate nodes

are found within the two-hop range, other methods should be considered to reconstruct the network, such as adding nodes of the same type, utilizing intermediate nodes for forwarding, and so on. This specific case is not discussed in this paper. Thus, we prioritize timely reconstruction measures for nodes with higher degrees to maximize the integrity of the remaining task chains.

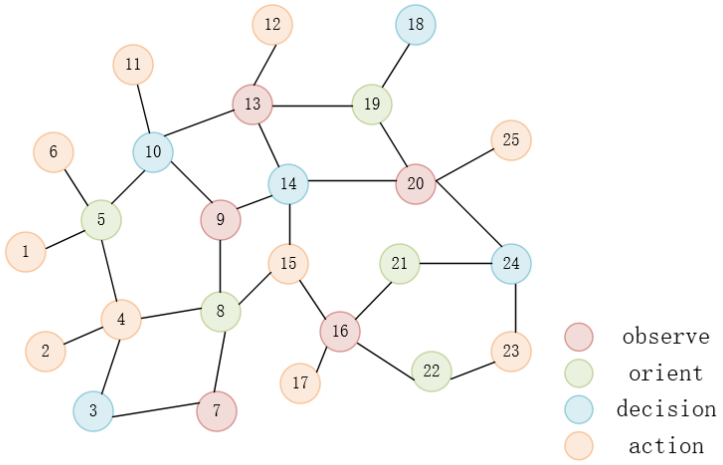


Figure 6. The topology structure of the UAV network.

Algorithm 1 Redistribute Data After Link Failure

procedure REDISTRIBUTEDATAAFTERLINKFAILURE(linkFailureList)
  Input: *linkFailureList* ▷ List of failed links with data and capacities
  Output: remaining *taskintegrity* ▷ Data redistribution after link failure
  Initialization:
    $\Gamma_i \leftarrow$ getNeighborNodes(linkFailure.node$_i$) ▷ Neighboring nodes of node v_i
    $\Gamma_j \leftarrow$ getNeighborNodes(linkFailure.node$_j$) ▷ Neighboring nodes of node v_j
  for each e_{ij} in linkFailureList **do**
    for each e_{ik} in Γ_i **do**
      $\Delta L_{ik} \leftarrow L_{ij} \times \frac{C_{ik}}{\sum_{a\in\Gamma_i} C_{ia}+\sum_{b\in\Gamma_j} C_{jb}}$ ▷ Calculate extra data allocation
      $L_{ik}^{\text{new}} \leftarrow L_{ik} + \Delta L_{ik}$ ▷ Update data carried on link e_{ik}
      if $L_{ik}^{\text{new}} > C_{ik}$ **then** ▷ Check for communication congestion
        handleCommunicationCongestion(e_{ik})
      end if
    end for
  end for
end procedure

5. Simulation Results

For the simulation, we consider an area of 5 × 5 km with 25 UAVs that are located uniformly in this area. As for the quantities of various types of UAVs, to avoid an excessive number of UAVs of the same type, we have decided to specify a range of values while still maintaining the total sum as a predetermined value. Additionally, based on the experimental findings in Ref. [25] and the various operational requirements for firefighting in real-life scenarios, we have determined that there will be a higher probability of having a greater number of UAVs dedicated to firefighting compared to other types of UAVs. The UAV can communicate with other devices with their communication range and move at a fixed speed in random directions within the scene range. We have used MATLAB (R2023a) for the simulation model, and the initial simulation parameters are given in Table 2.

Table 2. Parameters of the simulation.

Parameters	Values
Size of area	5 × 5 km
Time of iteration	2 s
Number of iterations	20
Number of Observe nodes	1–10
Number of Orient nodes	1–10
Number of Decide nodes	1–10
Number of Act nodes	5–10
Maximum UAV speed	20 m/s
Number of vulnerable nodes	5
Node mobility	Random way-point model

5.1. Vulnerability Analysis

Before conducting a vulnerability analysis on the network, we first need to determine the weights of various indicators in UAV communication attributes and functional topology attributes. It is not feasible to experiment with every possible value, so we compare network vulnerability performance under two types of weight proportions (emphasizing a certain indicator and relative balance) to assess the network's vulnerability.

In Figure 7, a series of experiments were conducted where the weights for node degree, node betweenness, and node closeness centrality in the communication topology were set as (0.2, 0.2, 0.6), (0.2, 0.6, 0.2), (0.6, 0.2, 0.2), and (0.3, 0.3, 0.4), (0.3, 0.4, 0.3), (0.4, 0.3, 0.3). Different node communication topology importance rankings were obtained based on these weight settings. Then, according to the ranking order, the most important nodes were iteratively removed from the current network, and the remaining task chain integrity was calculated. The experimental results show that, when there was a significant difference in weights among the three indicators, the rate of decline in network task chain integrity was relatively slow. It required removing 9 to 98 nodes to completely reduce it to 0. However, when the weights of the three indicators were close, the average decrease in remaining network connectivity after removing nodes was 0.13 compared to the previous group. Only 5 nodes needed to be removed to reduce the remaining task chain integrity of the network to 0. This suggests that weight allocation for communication topology attributes should be as evenly distributed as possible.

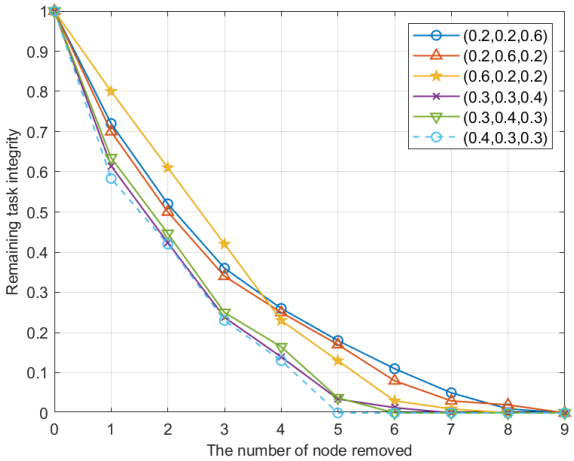


Figure 7. Effect of Topology Attribute Proportions on Vulnerability Analysis.

Similarly, in Figure 8, this experiment also set the weights for substitutability, connectivity importance, and efficiency importance in the functional topology as (0.2, 0.2, 0.6), (0.2, 0.6, 0.2), (0.6, 0.2, 0.2), and (0.3, 0.3, 0.4), (0.3, 0.4, 0.3), (0.4, 0.3, 0.3). Different UAV importance rankings were obtained based on these weight settings, and then the most important nodes were sequentially removed. By observing the results, it was found that,

when the allocation ratio was (0.3, 0.4, 0.3), the removal of a single node had the greatest impact on the remaining task chain integrity of the network, and only 4 nodes needed to be removed to reduce the task chain integrity of the network to 0. Therefore, this weight allocation ratio became the default proportion for subsequent experiments.

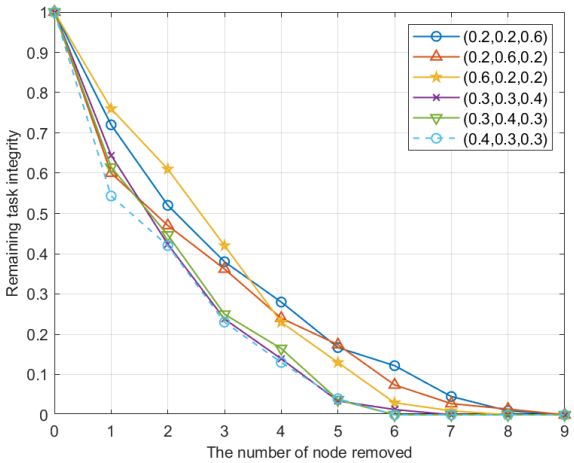


Figure 8. Effect of Functional Attribute Proportions on Vulnerability Analysis.

Finally, this experiment mainly deals with the impact of the ratio of functional topology attributes to communication topology attributes on the effectiveness of vulnerability analysis. Based on the results of the previous two experiments, this experiment sets the weight distribution of indicators within the communication topology attributes as (0.4, 0.3, 0.3) and the weight distribution of indicators within the functional topology attributes as (0.3, 0.4, 0.3). Then, by adjusting the weights of the two types of attributes, the importance ranking of nodes is obtained, and the most important nodes under the current weight distribution are removed to obtain the remaining task chain integrity of the network. In Figure 9, it can be observed that, when the weight coefficient of communication topology attributes ϕ_1 is 0.55, and when the weight of functional topology attributes ϕ_2 is 0.45, removing the most important nodes under this weight distribution causes the greatest decrease in network task chain integrity. Therefore, this weight allocation ratio is used for subsequent vulnerability analysis experiments.

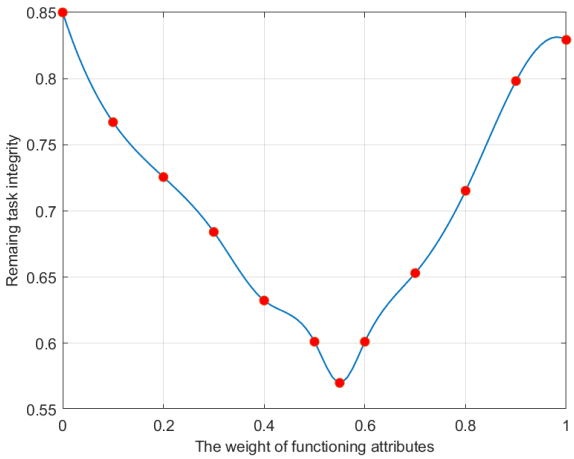


Figure 9. Vulnerability Analysis Effectiveness under Different Importance Weightings.

After confirming the weight values, in each iteration, we obtain the vulnerability node importance ranking through MENVAL. At the same time, we also use the traditional topological analysis method (NVA(Node Vulnerability Algorithm)), in which the selection of important nodes is completely based on the importance of communication topology.

For the UAV network topology depicted in Figure 6, the node importance rankings obtained through MENVAL and NVA are shown in Tables 3 and 4.

Table 3. Importance ranking through MENVAL.

Number	Node ID	$\beta_i^{'}$	$W_i^{'}$	$\rho_i^{'}$
1	9	0.68	0.76	0.72
2	14	0.70	0.63	0.67
3	10	0.69	0.64	0.66
4	8	0.68	0.60	0.64
5	20	0.65	0.59	0.62

Table 4. Importance ranking through NVA.

Number	Node ID	$\beta_i^{'}$
1	14	0.70
2	10	0.69
3	8	0.68
4	9	0.68
5	20	0.65

We sequentially attack the identified vulnerable nodes obtained from both methods by removing all communication links of each node. The vulnerability of nodes is then assessed by evaluating the integrity of the remaining task chains, and the simulation results after the attack are shown in Figure 10.

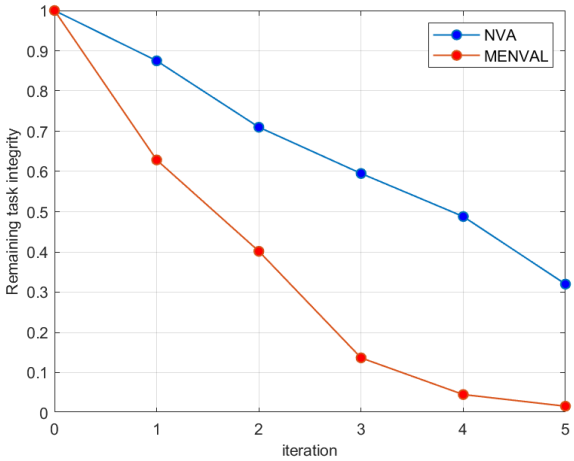


Figure 10. Different Algorithm-Derived Node Vulnerability Effects

The horizontal axis represents the number of iterations, and the vertical axis represents the remaining task chain integrity of the network. In each iteration, we remove a vulnerable node and then calculate the remaining task chain integrity in the network. In Figure 11, we can observe that, with the removal of each node, the network vulnerability decreases accordingly. However, it is evident that the removal of vulnerable nodes obtained by MENVAL has a significantly greater impact on the network. By the fifth iteration, the network connectivity is almost zero. In contrast, attacking the vulnerable nodes obtained by the NVA method results in a relatively slower decrease in network metrics, and the remaining task chain integrity at the end of the last iteration is also higher than that of the first method.

Then we set the node scale to 25, 50, 75, and 100, and the number of various types of nodes increases proportionally to the corresponding numbers. In each iteration, we remove vulnerable nodes obtained by MENVAL in order of importance and compare the impact of removing vulnerable nodes on the UAV network at different node scales.

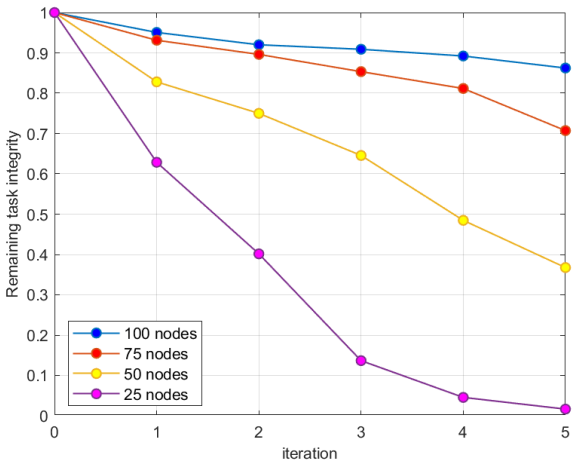


Figure 11. The network's remaining connectivity change curve on a multi-scale network.

In Figure 11, we can observe that, as the scale of the UAV network increases, the impact of removing individual key nodes on the remaining task connectivity in the network gradually decreases. This occurs when there are a larger number of UAVs and a greater number of task chains, where the removal of individual nodes has a weaker impact on the entire network. Conversely, in networks with fewer UAVs and a relatively smaller number of task chains, the impact of the removal of individual UAVs on the network becomes more pronounced.

5.2. Topology Reconstruction Strategy

Based on the vulnerability analysis results, we proceed with experiments related to the topological reconstruction algorithm NITRA. Taking the example of Node 9 failure on the generated topology network shown in Figure 6, we sort the importance of the one-hop range nodes from small to large in Table 5.

Table 5. Importance of the one-hop range nodes through NITRA.

Number	Node ID	Function	$\rho_i^{'}$
1	8	Orient	0.64
2	10	Decide	0.66
3	14	Decide	0.67

Within the one-hop range, there are no nodes of the same type as Node 9. Since Node 9 is a vulnerable node, according to the algorithm rules, we need to continue searching for candidate nodes set within the two-hop range, which is represented in Table 6.

According to Table 6, within the two-hop range, both Node 13 and Node 20 are of the same type as Node 9. Additionally, the comprehensive importance of Node 13 is less than that of Node 20 and Node 9. So, Node 13 is considered a candidate node for Node 9. Using the same method, candidate nodes for other vulnerable nodes can be determined.

At the same time, we also use the traditional topological reconstruction strategy MCMA, in which the selection of candidate nodes is completely based on the degree of the nodes. Obviously, by using MCMA, the candidate node for Node 9 is Node 8 because of its smallest degree and the lowest index number.

Table 6. Importance of the two-hop range nodes through NITRA.

Number	Node ID	Function	$\rho_i^{'}$
1	11	Act	0.64
2	7	Observe	0.66
3	5	Orient	0.67
4	4	Act	0.67
5	15	Act	0.68
6	13	Observe	0.70
7	20	Observe	0.71

We conducted 20 experiments to compare MCMA and NITRA, with parameters consistent with Table 2. Each experiment generated a different initial network topology, and a random vulnerable or non-vulnerable node was removed from the network. Subsequently, various topological reconstruction methods were applied to the network after removal. After the simulation concluded, we compared the integrity of the remaining task chain and the distance moved by the candidate UAVs under both algorithms. The average values of the evaluation metrics obtained by the two methods were then calculated. The experimental results are shown in Figures 12 and 13.

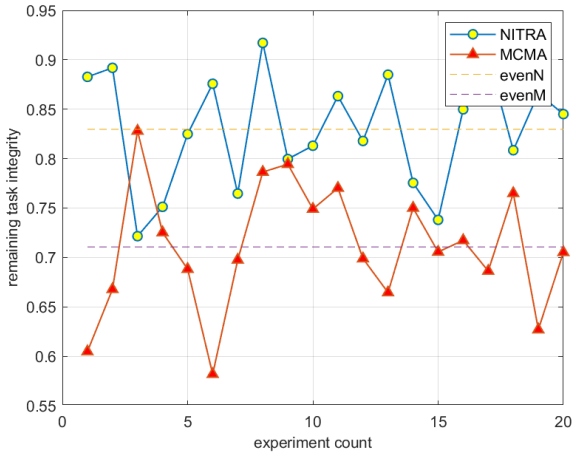


Figure 12. Removing a random vulnerable node in generated networks.

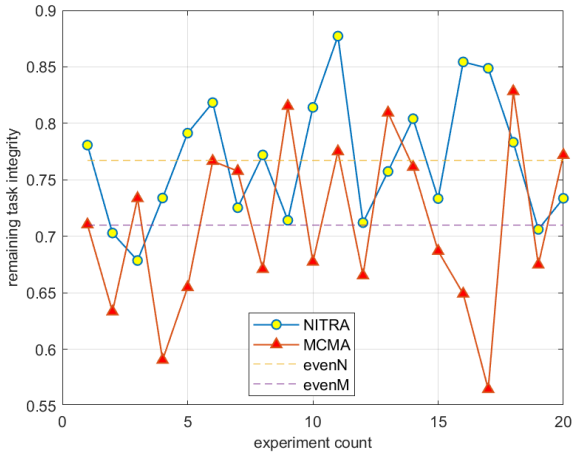


Figure 13. Removing a random non-vulnerable node in generated networks.

In Figure 12, each experiment randomly removes one vulnerable node from the generated network, and then different reconstruction algorithms are used to repair the network topology, represented by dashed lines indicating the averages. Through observation, it can be noted that, although in a few experiments the MCMA method slightly outperforms the proposed NITRA method in terms of remaining network connectivity, overall, networks reconstructed using NITRA are superior to those using MCMA. The residual task chain integrity of NITRA is 0.83, whereas the MCMA's average is 0.71, indicating a 17% improvement over MCMA.

In Figure 13, each experiment randomly removes a non-vulnerable node from the network. From the experimental results, it can be observed that, compared to the removal of vulnerable nodes, in overall performance, NITRA has an advantage in restoring network connectivity over MCMA. The average residual task chain integrity of networks reconstructed using NITRA is 0.77, while using MCMA for reconstruction yields a value of 0.70. Although the repair effect is not as pronounced as in the case of vulnerable node failure, it still represents a 10% improvement over MCMA. This indicates that, in ensuring the integrity of the remaining task chain during network reconstruction, NITRA is superior to MCMA, thus better maintaining network task execution performance. Both sets of experiments demonstrate that, in UAV task chain networks, NITRA surpasses MCMA in ensuring residual task chain integrity, thereby better restoring network task execution performance in the event of node failures.

In Figures 14 and 15, experiments were conducted to compare the movement distances of candidate UAVs after removing random nodes using two different reconstruction algorithms. It can be observed that, in most experiments, compared to MCMA, NITRA results in shorter average travel distances for candidate UAVs, reducing them by 26%. Particularly noteworthy are experiments 7 and 15 in Figure 15, where, because the algorithm did not find candidate nodes within the two-hop range of failed nodes, NITRA did not perform topology reconstruction. Although MCMA achieved higher residual task chain integrity relative to NITRA in experiment 7, the improvement was not significant. Even in experiment 15, the obtained residual task chain integrity was not as good as that of NITRA. This confirms that NITRA, compared to MCMA, can reduce the movement distance of candidate UAVs, thereby lowering the energy consumption of UAVs and prolonging the operational time of UAV networks.

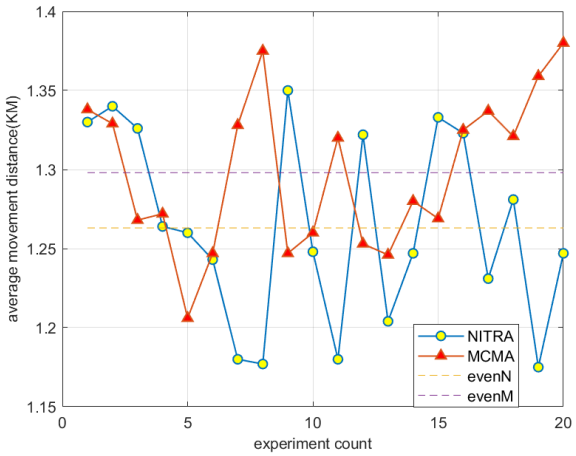


Figure 14. Average movement distance change curve upon removing a vulnerable node.

In Figure 16, we compare the time taken for network recovery to a steady state under different topology reconstruction strategies using NITRA and MCMA after node failures occur in the network. It can be observed that, within 0.6 s after node failure, the remaining task integrity of both algorithms rapidly decreases. After 0.6 s, the network under the NITRA strategy reduces its rate of decline and reaches a steady state around 1.7 s, with a remaining task integrity of approximately 0.75. The network under the MCMA

strategy declines more rapidly, reaching a steady state around 2.3 s, with a remaining task connectivity of 0.63. Therefore, NITRA can restore network stability faster than MCMA, maximize the remaining task integrity after node failures, and improve it by about 19%.

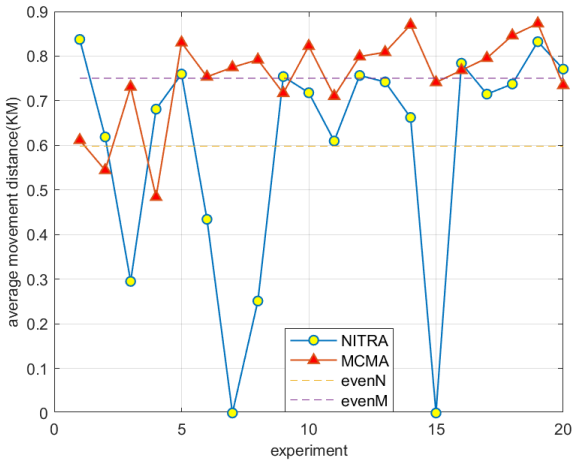


Figure 15. Average movement distance change curve upon removing a vulnerable node.

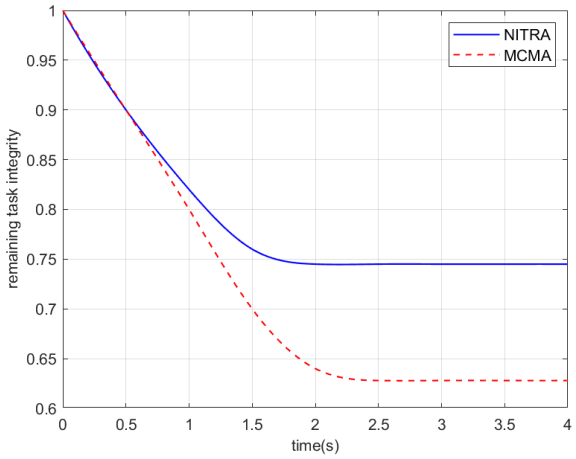


Figure 16. Average movement distance change curve upon removing a non-vulnerable node.

6. Conclusions

In this paper, we propose the VUTRM algorithm to address node failure issues in UAV task chains. The algorithm comprises two parts: MENVAL and NITRA. For MENVAL, an innovative multi-metric node evaluation method is employed to comprehensively consider both communication and functional importance of nodes, thereby enhancing the accuracy of ranking vulnerable nodes in the network. NITRA, based on the obtained node importance ranking, adopts different reconstruction strategies for vulnerable and non-vulnerable nodes. Experimental results indicate that, in small-scale networks, the vulnerable nodes identified by MENVAL have a more significant impact on the network task chains, necessitating targeted protection. Furthermore, in scenarios of single-node failure, compared to MCMA, NITRA demonstrates improvements in ensuring the connectivity of the remaining task chains and reducing the movement distance of candidate nodes, thus enhancing the robustness of UAV networks.

Author Contributions: Conceptualization, J.L. and Z.H.; methodology, Q.Y. (Qinghai Yang); software, Q.Y. (Qingfeng Yue); validation, X.X.; writing—original draft preparation, Q.Y. (Qinggfeng Yue); writing—review and editing, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Natural Science Basis Research Plan in the Shaanxi Province of China (2023JCYB555) and by the National Key Research and Development Program of China (2020YFB1807700).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Zijia Huang was employed by the company The 20th Research Institute of China Electronics Technology Group Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAVs	Unmanned Aerial Vehicles
VUTRM	Vulnerability-based Topology Reconstruction Mechanism
MENVAL	Multi-metric Node Vulnerability Assessment Algorithm
NITRA	Node Importance-based Topology Reconstruction Algorithm
MCDS	Minimum Connected Dominating Set
RIM	Resilient Information Matrix
DARA	Degree and Relative Distance Aware
PADRA	Partition-Aware Dominating Set-based Recovery Algorithm
LeDiR	Least Disturbance Routing
ND	Network Diffusion
NIO	Influence in the Loop
NVA	Node Vulnerability Algorithm
MCMA	Minimum Cascade Mobile Algorithm

References

1. Ding, G.; Wu, Q.; Zhang, L.; Lin, Y.; Tsiftsis, T.A.; Yao, Y.-D. An amateur drone surveillance system based on the cognitive internet of things. *IEEE Commun. Mag.* **2018**, *56*, 29–35. [CrossRef]
2. Sahu, A.; Davis, K. Structural learning techniques for bayesian attack graphs in cyber physical power systems. In Proceedings of the 2021 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 2–5 February 2021; pp. 1–6.
3. Liu, X.; Abdelhakim, M.; Krishnamurthy, P.; Tipper, D. Identifying malicious nodes in multihop iot networks using diversity and unsupervised learning. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
4. Allodi, L.; Banescu, S.; Femmer, H.; Beckers, K. Identifying relevant information cues for vulnerability assessment using cvss. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 19–21 March 2018; ser. CODASPY '18; Association for Computing Machinery: New York, NY, USA, 2018; pp. 119–126. [CrossRef]
5. Akkaya, K.; Thimmapuram, A.; Senel, F.; Uludag, S. Distributed recovery of actor failures in wireless sensor and actor networks. In Proceedings of the 2008 IEEE Wireless Communications and Networking Conference, Las Vegas, NV, USA, 31 March–3 April 2008; pp. 2480–2485.
6. Asim, M.; Mokhtar, H.; Khan, M.Z.; Merabti, M. A sensor relocation scheme for wireless sensor networks. In Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, Singapore, 22–25 March 2011; pp. 808–813.
7. Zhao, Z.; Li, Z.; Guo, H.; Yang, M. Research on vulnerability of space information network based on interdependent networks. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18–21 June 2018; pp. 302–309.
8. Asadpour, M.; Hummel, K.A.; Giustiniano, D.; Draskovic, S. Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 843–856. [CrossRef]
9. Li, B.; Fei, Z.; Zhang, Y.; Guizani, M. Secure uav communication networks over 5G. *IEEE Wirel. Commun.* **2019**, *26*, 114–120. [CrossRef]
10. Poulin, R.; Boily, M.-C.; Mâsse, B.R. Dynamical systems to define centrality in social networks. *Soc. Netw.* **2000**, *22*, 187–220. [CrossRef]
11. Howell, N. Applied network analysis: A methodological introduction. *Can. J. Sociol.* **1985**, *10*, 209–211. [CrossRef]
12. Freeman, L.C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1978**, *1*, 215–239. Available online: https://www.sciencedirect.com/science/article/pii/0378873378900217 (accessed on 22 May 2024). [CrossRef]
13. Zhao, M.; Wu, M.; Qian, L.; An, Q.; Li, S. Evaluation of cross-layer network vulnerability of power communication network based on multi-dimensional and multi-layer node importance analysis. *IEEE Access* **2022**, *10*, 67181–67197.

14. Dolev, S.; Elovici, Y.; Puzis, R. Routing betweenness centrality. *J. ACM* **2010**, *57*, 1–27. [CrossRef]
15. Osinga, F. Getting'a discourse on winning and losing: A primer on boyd's 'Theory of intellectual evolution'. *Contemp. Secur. Policy* **2013**, *34*, 603–624. [CrossRef]
16. Diaz, D.O. An Optimization-Based Approach to Measuring Robustness in Command and Control Networks. Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, USA, 2019.
17. Cao, Y.-Y.; Liu, R.-R.; Jia, C.-X.; Wang, B.-H. Percolation in multilayer complex networks with connectivity and interdependency topological structures. *Commun. Nonlinear Sci. Numer.* **2021**, *92*, 105492. [CrossRef]
18. Kenett, D.Y.; Gao, J.; Huang, X.; Shao, S.; Vodenska, I.; Buldyrev, S.V.; Paul, G.; Stanley, H.E.; Havlin, S. Network of interdependent networks: Overview of theory and applications. In *Networks of Networks: The Last Frontier of Complexity*; Springer: Cham, Switzerland, 2014; pp. 3–36.
19. Akram, V.K.; Ugurlu, O. Detecting the most vital articulation points in wireless multi-hop networks. *IEEE/ACM Trans. Netw.* **2023**, *31*, 2389–2402. [CrossRef]
20. Younis, M.; Lee, S.; Gupta, S.; Fisher, K. A localized self-healing algorithm for networks of moveable sensor nodes. In Proceedings of the IEEE GLOBECOM 2008–2008 IEEE Global Telecommunications Conference, New Orleans, LA, USA, 30 November–4 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–5.
21. Abbasi, A.A.; Younis, M.; Akkaya, K. Movement-assisted connectivity restoration in wireless sensor and actor networks. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *20*, 1366–1379. [CrossRef]
22. Senel, F.; Akkaya, K.; Younis, M. An efficient mechanism for establishing connectivity in wireless sensor and actor networks. In Proceedings of the IEEE GLOBECOM 2007—IEEE Global Telecommunications Conference, Washington, DC, USA, 26–30 November 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1129–1133.
23. Abbasi, A.; Younis, M.; Baroudi, U. Restoring connectivity in wireless sensor-actor networks with minimal topology changes. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–5.
24. Hao, W.; Wenbai, C.; Cui, H.; Hang, M. A minimum cascade mobile repair method for wireless sensor networks. *Microelectron. Comput.* **2021**, *38*, 31–37.
25. Wang, K.; Yuan, Y.; Chen, M.; Lou, Z.; Zhu, Z.; Li, R. A study of fire drone extinguishing system in high-rise buildings. *Fire* **2022**, *5*, 75. Available online: https://www.mdpi.com/2571-6255/5/3/75 (accessed on 22 May 2024). [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.