Hindawi
Security and Communication Networks
Volume 2021, Article ID 2009629, 16 pages
https://doi.org/10.1155/2021/2009629

Research Article

Optimal Network Destruction Strategy with Heterogeneous Cost under Cascading Failure Model

Fang Yang, Tao Ma, Tao Wu, Hong Shan, and Chunsheng Liu

College of Electronic Engineering, National University of Defense Technology, Hefei, Anhui 230037, China

Correspondence should be addressed to Tao Wu; terence.taowu@gmail.com

Received 26 June 2021; Revised 20 August 2021; Accepted 16 September 2021; Published 21 October 2021

Academic Editor: Konstantinos Fysarakis

By studying an attacker's strategy, defenders can better understand their own weaknesses and prepare a response to potential threats in advance. Recent studies on complex networks using the cascading failure model have revealed that removing critical nodes in the network will seriously threaten network security due to the cascading effect. The conventional strategy is to maximize the declining network performance by removing as few nodes as possible, but this ignores the difference in node removal costs and the impact of the removal order on network performance. Having considered all factors, including the cost heterogeneity and removal order of nodes, this paper proposes a destruction strategy that maximizes the declining network performance under a constraint based on the removal costs. First, we propose a heterogeneous cost model to describe the removal cost of each node. A hybrid directed simulated annealing and tabu search algorithm is then devised to determine the optimal sequence of nodes for removal. To speed up the search efficiency of the simulated annealing algorithm, this paper proposes an innovative directed disturbance strategy based on the average cost. After each annealing iteration, the tabu search algorithm is used to adjust the order of node removal. Finally, the effectiveness and convergence of the proposed algorithm are evaluated through extensive experiments on simulated and real networks. As the cost heterogeneity increases, we find that the impact of low-cost nodes on network security becomes larger.

1. Introduction

Complex networks can be used to describe a variety of interactive systems in social and natural environments, such as the Internet, power grids, transportation networks, and terrorist networks, [1–5]. Through detailed research into complex networks, Albert et al. [6] found that the removal of critical nodes will greatly affect the network performance. Motter and Lai [7] pointed out that because the network has a cascading failure phenomenon, intentional attacks can lead to a cascade of overload failures, which can in turn cause the entire or a substantial part of the network to collapse. As physical control systems are increasingly controlled by network-enabled devices, cyberattacks will have an important impact on the real world [8]. For example, the load frequency of the equipment in the power system is maliciously changed by remote programming, which further leads to the failure of the power system cascade [9, 10]; in 2012, part of the line in the Indian power system jumped, leading to the collapse of the northern power system [11], and network fluctuations at the autonomous system level have caused the Internet to collapse [12]. Thus, due to the redistribution of loads among nodes, component failure can lead to a cascade of overload failures, which can in turn cause all, or a substantial part, of the network to collapse.

Existing research has analyzed network vulnerability through the cascade model by removing critical nodes. Different node removal strategies will have completely different effects on network performance. For instance, Di Summa et al. [13] proposed an integer linear programming model with nonpolynomial constraints, which was linearly relaxed to solve the critical node selection problem in polynomial time, and an iterative two-phase algorithm has been developed to solve the cascading vulnerability node detection problem efficiently [14]. Seo et al. [15] proposed evaluation indicators for the cascading failure

phenomenon in a power system with the aim of minimizing the number of removed nodes while maximizing the declining network performance. These strategies are intended to minimize network performance by removing as few nodes as possible. In the aforementioned studies, the performance of the network can be described in terms of the pairwise connectivity (PWC) [13, 14] or the number of failed nodes [15].

However, existing research [13–15] does not consider the difference in node removal costs. Generally, more critical nodes will be protected more carefully and be more difficult to remove. For example, interdomain routing is protected by firewalls and intrusion detection systems that are difficult to permeate [16]—an analogy is that the cost of killing a core member of a terrorist network will be greater than that of killing other members [5]. This phenomenon of different node removal costs in the network is called cost heterogeneity. Simultaneously, the attackers are trapped by their own capabilities and are likely to have a limited attack budget. The attack budget is the maximum cost that an attacker can provide. As the cascading of the network is a load spreading process, different node removal sequences may interrupt this load spreading and have different effects on network performance (see Section 3.4).

Therefore, our problem is that of optimizing the network destruction strategy so as to maximize the declining network performance based on the cascade model when considering both the attacker's budget and the heterogeneous cost of node removal. First, as each node has a different removal cost, this paper proposes a heterogeneous cost model to describe the node removal cost. Generally, highly connected nodes play an essential role in real networks and are usually protected more carefully, so the cost of their removal is relatively high. Therefore, the heterogeneous cost model assumes that the node removal cost and degree are exponentially correlated and uses a cost-sensitive parameter γ for adjustment. Second, this article describes a hybrid directed simulated annealing and tabu search (DSA-TS) algorithm to search for the optimal node removal sequence. In each annealing iteration, the tabu search algorithm is used to adjust the order of node removal. Furthermore, in directed simulated annealing, an initial solution generation strategy and directional disturbance strategy are introduced to accelerate the convergence. Finally, this article uses network connectivity to evaluate the network performance. As the cost heterogeneity increases, low-cost nodes gradually threaten the security of networks. In summary, the main contributions of this work are as follows:

(i) The cascading failure and heterogeneous cost of nodes are modeled, and the node removal cost is found to be exponentially related to the node degree. Additionally, the difference in node removal order affects the spread of network cascading failures.

(ii) A hybrid DSA-TS heuristic algorithm is designed to find the optimal node removal sequence under a restricted attack budget. Extensive simulations and real network experiments indicate that DSA-TS is better than the baseline algorithm and achieves good convergence.

(iii) As the cost heterogeneity increases, more low-cost nodes begin to pose a serious threat to network security. Thus, the attack budget required to destroy the network entirely first increases and then decreases. From the perspective of the defender, this phenomenon shows that the defense strength should be appropriate for the importance of the nodes.

The remainder of this paper is organized as follows. Some related work is discussed in Section 2. In Section 3, a network model, cascading failure model, and heterogeneous cost model are proposed, and the problem of network destruction is defined. Section 4 describes the hybrid DSA-TS algorithm in detail. Section 5 analyzes the effectiveness and convergence of the algorithm through experiments on real and simulated networks. Section 6 determines the cascading characteristics and critical node characteristics of networks by analyzing the experimental phenomena. Finally, the conclusions to this study and ideas for future research are summarized in Section 7.

2. Related Work

Currently, the network destruction strategy is the classic problem of removing nodes in the network to maximize the declining network performance. The destruction of critical nodes will seriously threaten the security of the network in multiple fields, such as blockchain [17], big data [18], critical infrastructures [19], and IoT systems [20]. Due to the complexity of the network, a variety of efficient graph-based algorithms have been proposed [21–23]. In this section, we first survey the destruction strategy in a static network. Then, according the cascading failure characteristics of the network, the destruction strategy based on the cascading failure model is investigated.

Many destruction strategies have been proposed for static networks. The node sorting method sorts the nodes according to some evaluation index, typically based on structural characteristics of the network, such as the degree centrality [24], PageRank [25], or betweenness [26]. Nodes are removed according to on the sorting results, and indicators such as the number of remaining nodes in the network, largest component size [27], and network connectivity [28] are used to evaluate the network performance. This approach can be combined with specific application scenarios to evaluate the role of nodes. Liu et al. [29] combined complex network centrality theory and power system characteristics to give the electrical centrality, which can identify critical nodes in a power system. Another method is the node search strategy. By trying different node combinations, the nodes with the greatest impact on the network are selected. Because the solution space of this problem is huge, heuristic algorithms are often used to find a solution. Aringhieri et al. [30] proposed a local search metaheuristic algorithm that uses an iterative local search and a variable neighborhood search framework to disrupt the network by

deleting k nodes. Zhou et al. [31] used a variable population memetic search to solve the problem of selecting critical nodes in complex networks.

However, the aforementioned algorithms are only applicable in static network analysis. In the real world, the failure of one component in the network may cause other components to fail, such as in the power grid or in transportation and communication networks [11, 32, 33]. Zhao et al. [34] found that cascading failures can cause the network to become almost entirely disrupted. Therefore, we need to study the destruction strategy under the cascade model to detect cyber threats. Under the cascade model, the dynamic nature of the network makes it difficult to estimate the network performance after a node is removed. Various destruction strategies have been proposed to maximize the declining network performance. Yan et al. [35] used reinforcement learning to increase the damage of sequential topology attacks to the power network, while Zhang et al. [36] used a genetic algorithm to solve a multiobjective optimization problem under the cascading failure model and produced a variety of node selection schemes to provide attackers with choices. Zhu et al. [37] found that attacks based on load or degree are relatively ineffective and proposed a new attack strategy based on a risk graph. Wang et al. [38] pruned the solution space and used particle swarm optimization (PSO) to obtain the k critical line combinations in which faults caused the greatest damage to the power grid.

However, the above methods do not consider the cost of node heterogeneity and attackers with constrained budgets. For example, in a terrorist network [5], the layers of protection around core members make it much more challenging to kill terrorist leaders than other members. At the same time, the attacker's combat capability is limited in a complex area. Therefore, it is vital to design attack strategies based on one's own ability to disband and nullify terrorist groups as much as possible. We propose a hybrid DSA-TS algorithm to solve such problems. This method combines the simulated annealing algorithm and the tabu search algorithm and efficiently screens the optimal node removal sequence through the initial solution generation strategy based on node influence (CI) and the directional disturbance strategy.

3. Model and Problem Definition

In this section, we introduce the network model, cascading failure model, and heterogeneous cost model. We also define and analyze the complexity of the problem considered in this study.

3.1. Network Model. We model the network as an undirected graph $G = (V, E)$, where V is the node set and E is edge set in network G. The adjacency matrix $A(G) = (a_{ij})_{n\times n}$ represents the connection between nodes. If $a_{ij} = a_{ji} = 1$, there is an edge between nodes v_i and v_j; otherwise, there is no edge. We define the set of neighbors of node $u \in V$ as $N(u)$. In graph theory, the degree of a node $d(u)$ refers to the number of edges associated with the node. The degree of the node is equal to the number of neighbor nodes $d(u) = |N(u)|$.

3.2. Cascading Failure Model. Many cascading failure models [39–42] have been proposed. The local load redistribution model proposed by Wang et al. [42] is widely used to analyze applied flow networks such as power and communication systems. In our cascading failure model, each node has an initial load L_i and a processing load capacity C_i. When node v_i is destroyed, the load of the node will be offloaded to neighboring nodes according to the rules, so that neighbor node v_j will receive an increased load of ΔL_{ij}. If the node load exceeds its capacity $L_i + \Delta L_{ij} > C_i$, the node will be destroyed due to overloading. We define each parameter as follows.

Definition 1. In most networks, the node load is related to the degree. For simplicity, we define the node load L_i as a function of the degree $d(v_i)$. The initial load of node v_i is defined as

$$L_i = \alpha \times d(v_i)^p, \tag{1}$$

where α, p are used to control the strength of the correlation between the initial load and the node degree.

Definition 2. With the failure of a node, the increased load of the neighbor nodes is related to the degree. Obviously, a node with a higher degree can more easily receive the load. After node v_i is destroyed, the increased load ΔL_{ij} received by the neighbor node v_j is defined as

$$\Delta L_{ij} = L_i \times \frac{\mathrm{d}(v_j)}{\sum_{u\in N(v_i)} \mathrm{d}(u)}, \tag{2}$$

where $N(v_i)$ is the neighbor node set of node v_i.

Definition 3. In real networks, the capacity is severely limited by cost. The capacity of a node is assumed to be linearly related to the initial load of the node [7], defined as

$$C_i = \lambda \times L_i, \tag{3}$$

where λ is related to the initial load and characterizes the capability of a node.

Figure 1 shows a simple example of how the load is distributed when a node is removed. The initial network parameters are $\alpha = p = 1$ and $\lambda = 1.5$. According to these parameters and the node degree, we initialize the load and capacity of each node (see equations (1) and (3)).

3.3. Heterogeneous Cost Model. In different application scenarios, the node removal costs will typically be different. Without loss of generality, we believe that more important nodes will have a higher level of protection. The importance of nodes is usually determined according to the scenario, e.g., the transportation hub in a transportation network and core routes in a communication network. The cost of v_i is defined as c_i. In this paper, without loss of generality, we assume that the node removal cost is related to the degree of the node. As the

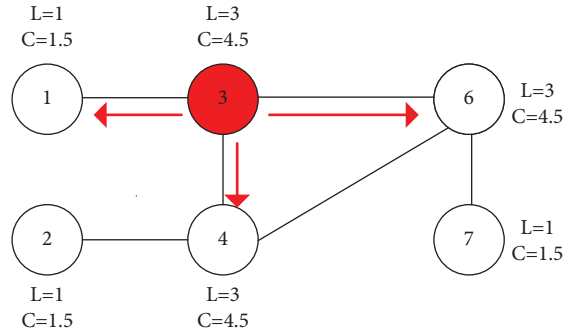


Figure 1: Simple load redistribution example. If node 3 is removed, its load will be distributed to neighboring nodes in accordance with the principle of load distribution (see equation (2)), such as $\Delta L_{31} = 3 \times 1/1 + 3 + 3 = 3/7$ and $\Delta L_{34} = \Delta L_{36} = 9/7$. Because $C_1 > L_1 + \Delta L_{31}$ and the load of nodes 4 and 6 does not exceed the capacity of the node, the network does not cascade.

degree of the nodes is moderately heterogeneous, the cost of node removal is also heterogeneous.

However, under the fixed overall network protection resources, the total cost of removing all nodes is fixed [43]. It is impossible to provide unlimited protection measures for a certain node. In this paper, we define the sum of the removal costs of all nodes in the network as the sum of the node degrees in the network, $c_s = \sum_{i=0}^{n} d(v_i)$. The removal cost of each node is related to the total cost of removing all nodes. Therefore, an increase in the node removal cost reflects not only an increase in the difficulty of the attack but also the greater level of protection of the node. At the same time, the attacker's budget B is constrained, which is related to the sum cost of removing all nodes in the network. Each parameter can be formulated as follows.

Definition 4. The cost of node v_i is defined as

$$c_i = \frac{d(v_i)^{\gamma}}{\sum_{i=0}^{n} d(v_i)^{\gamma}} \times c_s, \tag{4}$$

where $\gamma \geq 0$ is a cost-sensitive parameter. When $\gamma = 0$, the cost of each node is the same; as γ increases, the node costs become more heterogeneous.

Definition 5. The attacker's budget B is defined as

$$B = \beta \times c_s, \tag{5}$$

where $\beta \in [0, 1]$ is the budget constraint parameter, which describes the ability of the attacker. From the perspective of extremes, when $\beta = 0$, we cannot remove node from the network, and all nodes can be removed when $\beta = 1$.

3.4. Problem Description. Some system crashes are caused by a small number of critical nodes. However, different node removal sequences will have different effects on the network due to the cascading effect. In Figure 2, different removal methods and removal sequences have completely different effects on the network. As the transmission of network cascading is essentially load propagation, different removal sequences may interrupt or change the spread of the load. This article considers the impact on node-by-node removal on the network. After the current node removal completes the network cascade, the next node is removed. We define the removal sequence as $\mathrm{DS} = \{v_1, v_2, \ldots, v_k\}$. The removal cost of each node is different, and more important nodes tend to have higher removal costs (see equation (4)). Therefore, in the case of cost heterogeneity under a constrained removal cost, the goal of this research is to choose a node removal sequence DS that minimizes the network performance degradation rate $F(\mathrm{DS})$. We define the problem as follows.

Definition 6 (budget-constrained network destruction (BCND) problem). Given a network $G = (\mathrm{V}, \mathrm{E})$ and the attacker's budget B, choose a suitable node removal sequence $\mathrm{DS} \in V$ so as to minimize the network performance degradation rate by cascading. BCND is formally defined as follows.

$$\begin{aligned} &\text{Minimize } F\left(\mathrm{DS} = \{v_1, v_2, \ldots, v_k\}\right) \\ &s.t. \quad c_{\mathrm{DS}} \leq B, \mathrm{DS} \in V, \end{aligned} \tag{6}$$

where c_{DS} is the total cost of node set DS. F is our optimization goal, defined as the ratio between the remaining network and the original network after the network is attacked, $F(DS) = \Gamma(G/DS)/\Gamma(G)$.

For different application scenarios, our network evaluation methods vary. Without loss of generality, the topological characteristics of the network are often used as indicators for evaluating network performance, such as the number of remaining nodes in the network, the largest component size [27], and the network connectivity [28]. Here, we use the network connectivity as an index to evaluate network performance. Network connectivity describes the connectivity between any two points in the network.

Definition 7. Network connectivity can be defined as

$$\Gamma(G) = \sum_{g_i \in G} \frac{\delta_i \cdot (\delta_i - 1)}{2}, \tag{7}$$

where g_i is a connected subgraph in network G and δ_i is the number of nodes in connected subgraph g_i.

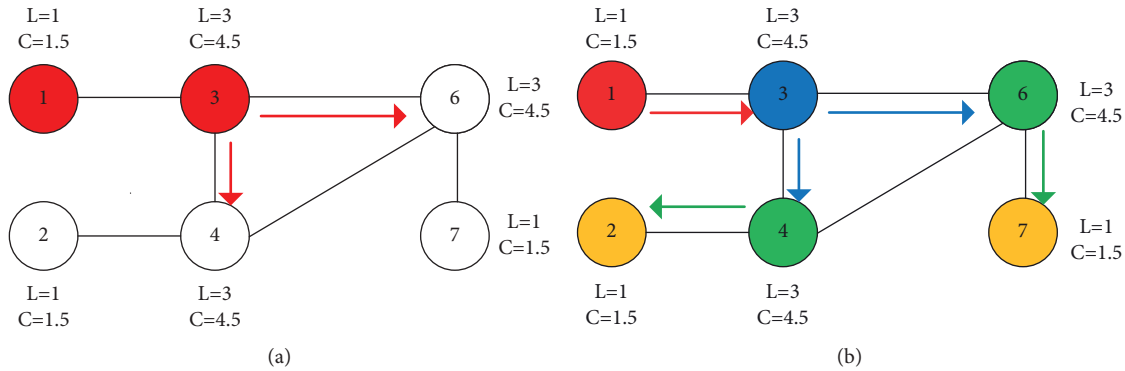


Figure 2: Example describing the impact of different types of removal method on the network. The network parameters are consistent with Figure 1. In this figure, (a) represents directly removing nodes 1 and 3 and (b) represents removing nodes 1 and 3 in order. Obviously, as nodes 1 and 3 are directly attacked, the load of node 1 cannot be transmitted to node 3, and the network does not cascade in (a). In (b), due to the sequential removal of nodes 1 and 3, all nodes in the network fail.

Usually, the search for an optimal removal set in the network is NP-hard. This shows that it is difficult to accurately solve this kind of problem in large-scale networks. Suppose that G is an undirected graph with n nodes. Although the cost is constrained, we need to compare approximately $n!$ solutions to obtain the optimal solution using an exhaustive method.

4. The Proposed Method (DSA-TS)

According to Definition 5, the large solution space makes it difficult to find an accurate solution to the BCND problem. At the same time, as pointed out in the Introduction, the problems of heterogeneous node cost and the different impact of the sequence of node removal on the network persist. To solve these problems, this paper proposes a hybrid DSA-TS algorithm. This section introduces the DSA-TS algorithm and describes the algorithm framework, initial solution generation strategy, directed simulated annealing algorithm, and tabu search.

4.1. Algorithm Framework. The overall framework of the algorithm is shown in Figure 3 and Algorithm 1. There are three main parts: initial solution generation strategy, directed simulated annealing algorithm, and tabu search. The main idea is to search for possible node combinations through the directed simulated annealing algorithm based on the initial solution. As different node removal sequences have different cascading effects, the tabu search algorithm and the simulated annealing algorithm are merged. Before each temperature drop, the tabu search algorithm is used to select the final sequence of node removal in the solution.

In Section 4.2, we introduce the initial solution generation strategy, which quantifies the influence of removing each node from the network. Algorithm 2 focuses on the problem whereby the initial solution is not necessarily the global optimum and proposes a directed simulated annealing algorithm. We improve the algorithm and propose a directional disturbance strategy, which mainly solves the low disturbance accuracy of the standard simulated annealing algorithm. Algorithm 3 considers different attacks that may cause different network cascading phenomena. A tabu search strategy is proposed to adjust the order of node removal. The parameters used in this paper are listed in Table 1.

4.2. Initial Solution Generation Strategy. The initial solutions of heuristic algorithms affect the generation of the final solution and the convergence speed of the algorithm. Usually, these initial solutions are generated at random. In the BCND problem, where the solution space is vast, a good initial solution will accelerate the convergence of the algorithm. Therefore, to speed up the convergence of the algorithm, we use the cascading potential [30] to design an index for evaluating the node values (node cascading influence, CI) and use this to generate the initial solution. The value CI_i/c_i is calculated under a unit cost for each node, and the results are sorted in descending order. The node removal sequence DS that produces the maximum node cost is selected from front to back as the initial sequence for the removal of nodes. The CI indicator evaluates the impact of removing a node on neighboring nodes and considers the importance of the removed node itself in the network structure.

Definition 8. CI is defined as follows:

$$CI_i = f\left(d_j\right) \times \left(\left|\Phi\left(i\right) \cup v_i\right| + \sum_{j \in N\left(i\right)/\Phi\left(i\right)} \frac{\Delta L_{ij}}{C_j - L_j} \right), \tag{8}$$

$$f\left(x\right) = \frac{1}{1 + e^{-x}},$$

where $\Phi(i)$ is the set of neighbor nodes that fail due to overload after v_i is removed. The term $\sum_{j \in N(i)/\Phi(i)} \Delta L_{ij}/C_j - L_j$ indicates that as the node is removed, some neighbor nodes do not fail, but their load increases. This factor must be considered because it makes it easier for the surrounding nodes to reach the state of being on the verge of overload, enhancing the influence of the removed node relative to

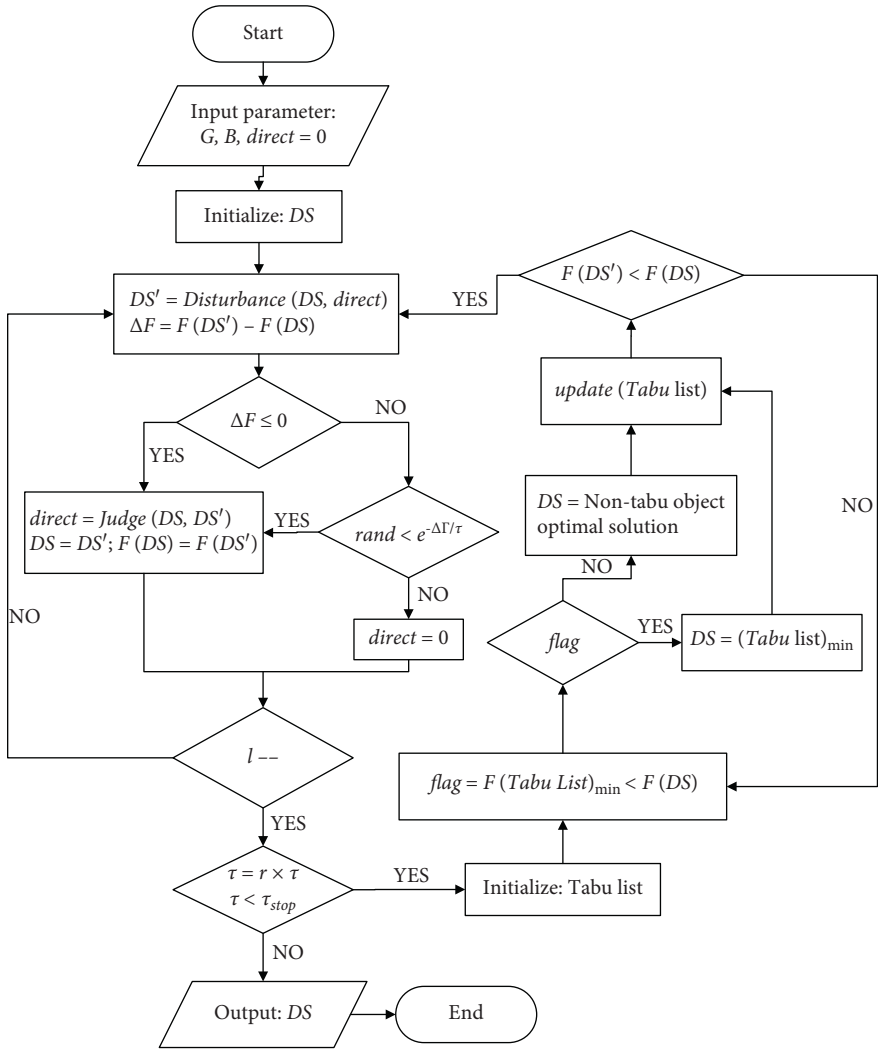


Figure 3: DSA-TS algorithm framework diagram.

Input: G: complex network; α, p: parameters controlling the correlation strength between load and degree; β: node redundancy parameter; γ: cost-sensitive parameter; c_s: total removal cost; τ: simulated annealing initial temperature; τ_{stop}: simulated annealing stop temperature; r: simulated annealing temperature reduction coefficient;
Output: the final disruption node set under the cost constraint DS;

(1) *Step 1*. Initialize the network.
(2) $G \longleftarrow Initialization(G, \alpha, p, \beta, \gamma)$
(3) $B = \beta \times c_s$
(4) *Step 2*. Generate initial solution.
(5) $DS \longleftarrow \text{InitialSolution}(G, CI, B)$
(6) *Step 3*. Directed simulated annealing algorithm.
(7) direct = 0
(8) while $\tau > \tau_{\text{stop}}$ do
(9) $\quad DS \longleftarrow \text{DSA}(G, \text{DS}, B, \text{direct})$
(10) $\quad$ *Step 4*. Tabu search algorithm.
(11) $\quad DS \longleftarrow \text{TS}(G, DS)$
(12) $\quad \tau = r \cdot \tau$
(13) end while
(14) return DS

Algorithm 1: DSA-TS.

Table 1: Parameters used in this paper.

Parameters	Description
G	Complex network
α, p	Parameters used to control the correlation strength between load and degree
β	Attack cost constraint parameters
c_s	Total cost of removing all nodes in the network
γ	Cost-sensitive parameter
λ	Node capacity redundancy parameter
τ	Simulated annealing initial temperature
τ_{stop}	Simulated annealing stop temperature
r	Simulated annealing temperature reduction coefficient
u_{sa}	Simulated annealing disturbance ratio
l	Simulated annealing number of inner cycles
direct	Simulated annealing disturbance direction
u_{ts}	Tabu search disturbance ratio
ε	Tabu length-related parameters

Input: u_{sa}: simulated annealing disturbance proportion; l: simulated annealing number of inner cycles;
Output: the node set within budget $DS_{\min}$;
(1) $DS_{\min} = DS$
(2) while l do
(3) *Step 1*. Directed disturbance strategy.
(4) if $direct == -1$ then
(5) ExtractList $\longleftarrow$ RandomMaxCost$(DS, u \cdot |DS|, c_{mean})$
(6) AddList $\longleftarrow$ RandomMinCost$(\overline{DS}, c_{mean})$
(7) $DS' \longleftarrow Exchange$(ExtractList, AddList, B)
(8) end if
(9) if $direct == 1$ then
(10) ExtractList $\longleftarrow$ RandomMinCost$(DS, u \cdot |DS|, c_{mean})$
(11) AddList $\longleftarrow$ RandomMaxCost$(\overline{DS}, c_{mean})$
(12) $DS' \longleftarrow Exchange$(ExtractList, AddList, B)
(13) else
(14) $Sorte\ dC\ I \longleftarrow Sorte\ d(\overline{DS}, CI)$
(15) $DS' \longleftarrow Exchange(ran\ do\ m(DS, u \cdot |DS|), Sorte\ dC\ I)$
(16) end if
(17) *Step 2*. Analyze the disturbance direction.
(18) if $c(DS')/|DS'| < c(DS_{\min})/|DS_{\min}|$ then
(19) $direct = -1$
(20) end if
(21) if $c(DS')/|DS'| > c(DS_{\min})/|DS_{\min}|$ then
(22) $di\ rect = 1$
(23) else
(24) $direct = 0$
(25) end if
(26) *Step 3*. Metropolis guidelines.
(27) if $F(DS') \le F(DS_{\min})$ then
(28) $DS, DS_{\min} = DS'$
(29) else
(30) if $random < e^{(-(\Gamma(G,DS') - \Gamma(G,DS_{\min}))/\tau)}$ then
(31) $DS = DS'$
(32) else
(33) $direct = 0$
(34) end if
(35) end if
(36) $l = l - 1$
(37) end while
(38) return $DS_{\min}$

Algorithm 2: DSA$(G, DS, B, direct)$.

Input: u_{ts}: tabu search disturbance ratio; ε: tabu length-related parameters;
Output: the final node set within budget $DS_{\min}$
(1) while $\lfloor 1/u_{ts} \rfloor$ − − do
(2) $TabuTable \leftarrow InitiTabuTable(DS)$
(3) end while
(4) while $\lfloor 1/u_{ts} \rfloor$ − − or $F(DS') > F(DS)$ do
(5) for i to $|TabuTable|$ do
(6) if $TabuTable[i].length > 0$ then
(7) $TabuTable[i].length$ − −
(8) $Candidatelist \leftarrow Candidate(DS)$
(9) end if
(10) end for
(11) if $F(DS') > F(DS)$ (DS' *do esnotcontaintabuelement* d) and $F(DS')_{\max} < F(DS)$ then
(12) $TabuTable[DS'_{\min}].length = \varepsilon \times \lfloor 1/u_{ts} \rfloor$
(13) $DS \leftarrow DS'_{\max}$
(14) else
(15) $TabuTable[DS'].length = \varepsilon \times \lfloor 1/u_{ts} \rfloor$
(16) (DS' *do esnotcontaintabuelement* d)
(17) $DS \leftarrow DS'$
(18) end if
(19) end while
(20) return DS

Algorithm 3: TS(G, DS).

other nodes. Simultaneously, the removal of high-level nodes may have a greater impact on the network, so we need to consider the effect on the network after high-level nodes are removed. Therefore, a monotonically increasing function $f(x)$ is introduced $(1/2 \le f(x) \le 1)$. A removed node with a higher degree will have a higher value of $f(x)$.

4.3. Directed Simulated Annealing Algorithm. The simulated annealing algorithm was proposed by Kirkpatrick et al. [44] in 1983. The main idea is to approximate the global optimum from the local optimum by simulating the annealing principle in metallurgical processing. "Annealing" is a physical term that refers to the process of heating and then cooling. The Metropolis criterion [45] is a key part of simulated annealing algorithms, as new states are determined probabilistically rather than completely deterministically. The Metropolis algorithm is the basis of simulated annealing, but its optimization speed is too slow when used directly. Therefore, to ensure convergence within a limited time, a new system was constructed in which the main parameters are the initial temperature τ and the end temperature τ_{stop}. This article uses the widely accepted geometric reduction law for the cooling step:

$$\tau_{k+1} = r \cdot \tau_k, \tag{9}$$

where τ_k represents the temperature of the algorithm after k iterations and r is the annealing rate, generally a constant value between 0.5 and 0.99.

The specific steps of the proposed algorithm are shown in Algorithm 2. The basic idea of the algorithm is to constantly disturb the initial solution DS to find the optimal node set of the network. However, the BCND problem has the characteristics of a constrained budget and large search space. If the random disturbance strategy generates a neighborhood solution based on the initial solution, the efficiency will be very low. To solve these two problems, this paper proposes a directed disturbance strategy, which can efficiently search for possible neighborhood solutions. Because the attack cost of each node is heterogeneous, the average cost of selecting different node combinations for removal is different. Therefore, we define the algorithm disturbance direction, *di rect*, to describe the disturbance direction of the neighborhood solution.

We assume that the initial disturbance direction, direct = 0, creates a random disturbance. $\overline{DS}$ refers to the unselected nodes in the network. When direct = 0, the algorithm randomly removes $u \cdot |DS|$ nodes and exchanges them with high-CI nodes in $\overline{DS}$ to generate DS'. If the disturbed node combination is better than the old node set, the direction of the disturbance depends on the change in the cost of the optimal solution relative to the suboptimal solution. When the average cost of the optimal solution is less than the suboptimal solution's average cost, *di rect* = −1 and the algorithm disturbs the node removal sequence in the direction of lower cost. Thus, the high-cost nodes in DS will be selected and exchanged with the low-cost nodes in DS'. When *di rect* = 1, the algorithm disturbs the node removal sequence in the direction of higher cost, and the low-cost nodes in DS are exchanged with the high-cost nodes in DS'. When the average removal cost is equal, *di rect* = 0 and a random disturbance is created. If the effect of the disturbed node combination is poor, the latest solution is accepted with probability $esp(-(\Gamma(G, DS') - \Gamma(G, DS_{\min}))/\tau)$. If the latest solution is accepted, the disturbance continues in the original direction; otherwise, a random disturbance is performed.

4.4. Tabu Search. The tabu search algorithm is a global step-by-step optimization algorithm based on a local neighborhood search. The algorithm has a fast convergence speed and can avoid becoming trapped around local optima. The principle was first proposed by Glover [46] in the late 1970s. After some development and improvement, a complete set of algorithms was finally formed. The algorithm used in this study is an improved tabu search algorithm. It aims to solve a problem with the original tabu search algorithm whereby combinatorial optimization becomes difficult under large-scale and restricted conditions. The designed algorithm can efficiently solve the BCND problem.

To solve the problem of the order of node removal causing the network to produce different cascading phenomena, we designed an internal search strategy based on the solution DS generated after the external search. This strategy changes the order in which nodes are removed to obtain the optimal collapse of the network (see Algorithm 3). The algorithm mainly comprises the following parts: movement mechanism, tabu table, amnesty rules, and termination criteria. We will introduce these parts in detail below.

The movement mechanism represents the process of the current solution moving to another solution, which determines the form of the solution generated in the neighborhood and the relationship between successive solutions. Therefore, a good movement mechanism will impact the search efficiency. Therefore, we regard the exchange of positions between nodes as a movement and introduce the internal exchange rate χ. Each exchange process involves $u_{ts} \cdot |DS|$ nodes.

The tabu list is a unique component at the core of the tabu search algorithm. It records and prohibits changes to prevent search loops from appearing and preventing the algorithm from becoming trapped around local optima. The critical factors for its design are the tabu object and the length of the tabu. The object and length of the tabu significantly affect the search speed and the quality of the settlement. Tabu objects are those limited by the tabu table. When initializing the tabu table, this algorithm selects $u_{ts} \cdot |DS|$ nodes from the initial solution DS and exchanges them with other unselected nodes. This exchange method reduces the chance of important nodes (in the order of removal) from being moved to the end. The tabu length is the number of iterations after which the tabu object fails. This paper introduces a tabu length parameter ε. The tabu length is related to the length of the tabu table $\varepsilon \cdot \lfloor 1/u_{ts} \rfloor$, where the value of ε is determined by the network size and experience.

The amnesty rule is a moderate relaxation of the tabu list. When a tabu object becomes the historical best solution, it is amnestied without being restricted by the tabu list. As the termination criterion, we use the length $\lfloor 1/u_{ts} \rfloor$ of the tabu table as the maximum number of iterations. If the obtained solution persistently exceeds the historical optimal solution, the algorithm continues even if the maximum number of iterations has been exceeded.

5. Experiments and Algorithm Analysis

In this section, we first introduce the experimental dataset and the comparison algorithm and then demonstrate the effectiveness of the proposed algorithm and the comparison algorithm on a simulated network and a real network under different cost heterogeneities. Further, we analyze the convergence of the algorithm when removing different components.

5.1. Experimental Setting

5.1.1. Experimental Parameters. We used Python 3.6 to run the simulations of scale-free networks on a PC with an Intel Core i7-9750 3.2 GHz CPU and 8.0 GB of RAM. The parameters of each part of the algorithm were set as follows: DSA parameters—initial temperature $\tau = 100$, termination temperature $\tau_{stop} = 1$, annealing rate $c = 0.8$, disturbance ratio $u_{sa} = 0.1$, and number of iterations $l = 10$; TS parameters—disturbance ratio $u_{ts} = 0.1$ and tabu length $\varepsilon = 0.2$.

5.1.2. Data Description. We first verify the effectiveness of the algorithm using synthetic and real networks. In this paper, three types of simulation networks with $N = 1000$ and $\langle k \rangle = 6$ are generated for experimentation.

Scale-free network generated by the Barabasi–Albert (BA) model [47]: the characteristic of scale-free networks is that a small number of nodes have a large number of connections and most other nodes have very few connections with a power-law degree distribution.

Small-world network generated by the Watts–Strogatz (WS) model [48]: in this kind of network, most arbitrary nodes can visit other nodes with fewer steps or hops.

Random network generated by the Erdös–Rényi (ER) model [49]: we connect each pair of nodes with a probability $p = 0.006$. Since each pair of nodes is connected with equal probability, the random network is a homogeneous network in which most of the nodes' degrees are around pN.

These three network models basically cover the complex network structure characteristics in reality.

As real networks, we consider six real network from an industry perspective, including power grids [50], a communication network [51], a road network [50], an interpersonal network [50], Facebook [52], and an economic network [50]. Different types of networks have different structural characteristics, such as different degree distributions, and different network sparseness. This feature affects the heterogeneity of costs and the robustness of the network. The purpose of the simulation network is to verify the universality of the algorithm for a certain type of network, and the real network verifies the validity of the application in reality. The specific characteristics of the networks are summarized in Table 2.

Table 2: Basic statistical characteristics of real networks.

Networks	n	m	$\langle k \rangle$	k_{max}	L	CC
Power grid	19	6637	2.66	19	18.56	0.11
Communication network	553	8979	4.66	553	3.56	0.37
Road network	1200	1400	2.413	10	18.37	0.02
Interpersonal network	416	2771	13.32	50	3.63	0.46
Facebook	4039	8823	443.691	523	3.69	0.61
Economic network	1300	7600	12	206	3.574	0.06

The real complex network structure parameters are shown in Table 2, including the number of nodes n and links m within the networks, the average degree $\langle k \rangle$ [24], the maximum degree k_{max}, the average shortest path length L, and the clustering coefficient CC [53] which represents the degree of clustering between nodes in a network.

5.1.3. Comparison Algorithm. To show the effectiveness of the proposed DSA-TS algorithm, we compare it with seven popular baseline algorithms, including HD, RIF, and HCI.

HD [24]. The HD algorithm sorts all nodes in the network by degree. On the premise of not exceeding a given cost, the node removal sequence runs from the largest to the smallest degree.

RIF [54]. This is the failure risk index, which calculates the ratio of the load of the node to the load of neighboring nodes: $RIF = L_i / \sum_{j \in N(i)} L_j$. A higher ratio indicates that the removal of the node is more likely to cause the failure of neighbor nodes. All nodes are sorted according to the RIF size, and the attack nodes are selected in order from largest to smallest without exceeding a given cost.

HCI. The HCI algorithm is sorted in descending order of each node's CI (Section 4.2), and the node removal sequence is determined from largest to smallest such that the attack meets the given cost constraints.

5.2. Algorithm Analysis

5.2.1. Effectiveness of the Proposed DSA-TS Algorithm. To verify the effectiveness of the proposed DSA-TS algorithm, its performance was compared with that of the baseline algorithms on simulated and real networks. First, a scale-free network, a small-world network, and a random network were generated with $n = 1000$ nodes and an average degree of $\langle k \rangle = 6$. The network load and capacity initialization parameters were set to $\alpha = p = 1$ and $\lambda = 2$. We compared the network connectivity under different removal costs. As shown in Figure 4, the DSA-TS algorithm achieves superior performance with different types of simulation networks and node cost heterogeneities. Note that when the cost-sensitive parameter $\gamma = 2$, the budget required to completely destroy the network based on degree (HD) and node influence (HCI) becomes progressively worse. In Figure 4(g), the network connectivity under the two removal strategies of HD and HCI suffers almost no drop. Therefore, as the cost heterogeneity increases, one cannot only consider the role of nodes and ignore the removal cost.

For the real networks, we considered six different domains and network characteristics for experimentation. The network load and redundant initialization parameters were consistent with those in Figure 4. As the effectiveness of the algorithm under different costs has been proved in Figure 4, this experiment mainly verified the effectiveness of the algorithm when the cost-sensitive parameter $\gamma = 1$. As shown in Figure 5, the proposed algorithm still achieves superior performance.

5.2.2. Convergence of DSA-TS. The proposed algorithm combines directed simulated annealing with a tabu search to solve the BCND problem. The role of each component in the algorithm is now examined. We deleted some of the algorithm components and compared the convergence with the complete algorithm under different budgets. The experimental results are shown in Figure 6, where DSA indicates the absence of the tabu search algorithm and SA-TS indicates the absence of the directed disturbance strategy. The abscissa is the number of algorithm iterations, and the ordinate is the minimum connectivity of the network. The experimental results show that the DSA-TS algorithm converges faster under different budgets and is less likely to become trapped around local optima, thus producing better results. By analyzing the experimental phenomena, it can be found that in Figure 6(a), the small removal cost means that the number of nodes that can be selected is limited and the algorithm convergence is similar, so no real difference is apparent. In Figures 6(b) and 6(c), the DSA-TS algorithm outperforms the comparison method in terms of convergence speed and optimal solution. By comparison, the directed search strategy has a significant impact on the algorithm and produces different results from the other two algorithms with $\beta = 0.15$. As the cost increases, the gap between DSA and DSA-TS becomes progressively smaller because as the removal cost increases, it becomes easier to cascade the network. Therefore, the DSA-TS algorithm achieves better performance in terms of convergence speed and convergence effect.

6. Experimental Results and Discussion

This section analyzes the cascade features and node removal characteristics of the networks under different heterogeneous costs. Since most networks in the real-world present scale-free characteristics, this section uses scale-free network with $N = 1000$ and $\langle k \rangle = 6$ for experimentation. We find that as the cost heterogeneity increases, low-cost nodes play an increasingly important role in network security.

6.1. Network Cascading Characteristics under Cost Heterogeneity. This section mainly examines the network

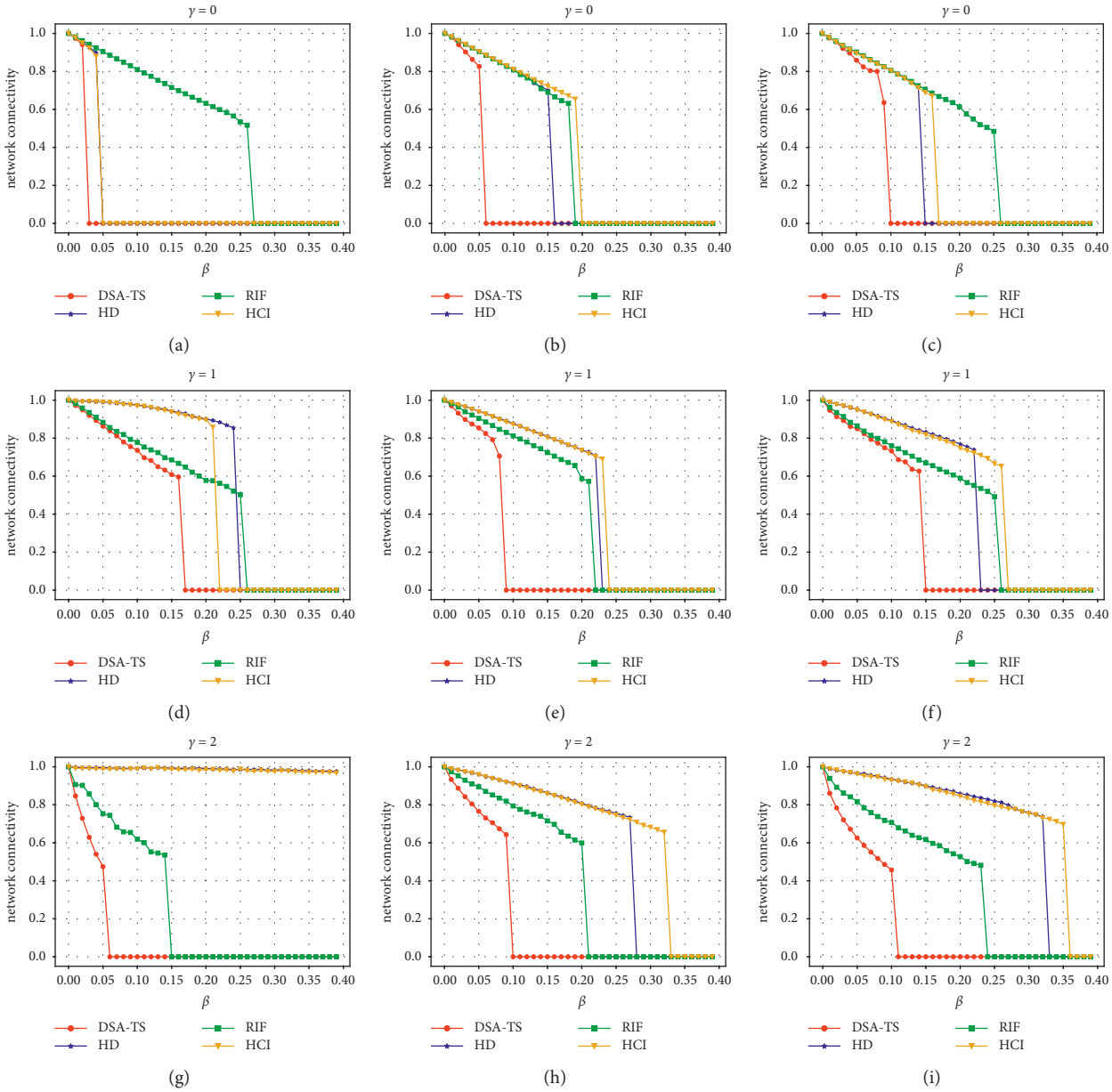


FIGURE 4: Effectiveness of DSA-TS applied to simulated networks under different cost heterogeneities. The experimental network parameters are $n = 1000$, $\widetilde{d} = 6$, $\alpha = p = 1$, and $\lambda = 2$. (a)–(c) $\gamma = 0$, (d)–(f) $\gamma = 1$, and (g)–(i) $\gamma = 2$. (a) SF network. (b) SW network. (c) ER network. (d) SF network. (e) SW network. (f) ER network. (g) SF network. (h) SW network. (i) ER network.

cascading characteristics under cost heterogeneity. As shown in Figures 3 and4, as the removal cost increases, the cascading of networks under different cost-sensitive parameters exhibits burstiness and unpredictability. Although the budget for complete network cascading decreases as the level of node redundancy increases, the generation of network cascading can still appear suddenly. In Figure 6, the abscissa is the removal cost parameter and the ordinate is the number of nodes in the network that have failed due to cascading. This feature increases the difficulty of algorithm design.

The budget required for complete network cascading under different cost-sensitive parameters is shown in Figure 7, where the abscissa is the cost-sensitive parameter and the ordinate is the cascading cost constraint parameter generated by the network. Here, we refer to the cost constraint parameter as the generation time of network cascading. As the total removal cost for a certain network is c_{sum}, earlier cascading generates a smaller budget B. This experiment shows that, under different node capacity redundancy parameters, an increase in cost heterogeneity causes the budget required for complete network cascading to first increase and then decrease and reach a peak around $\gamma = 1$. From the perspective of network defense, this phenomenon shows that defending too many or too few important nodes increases the network's

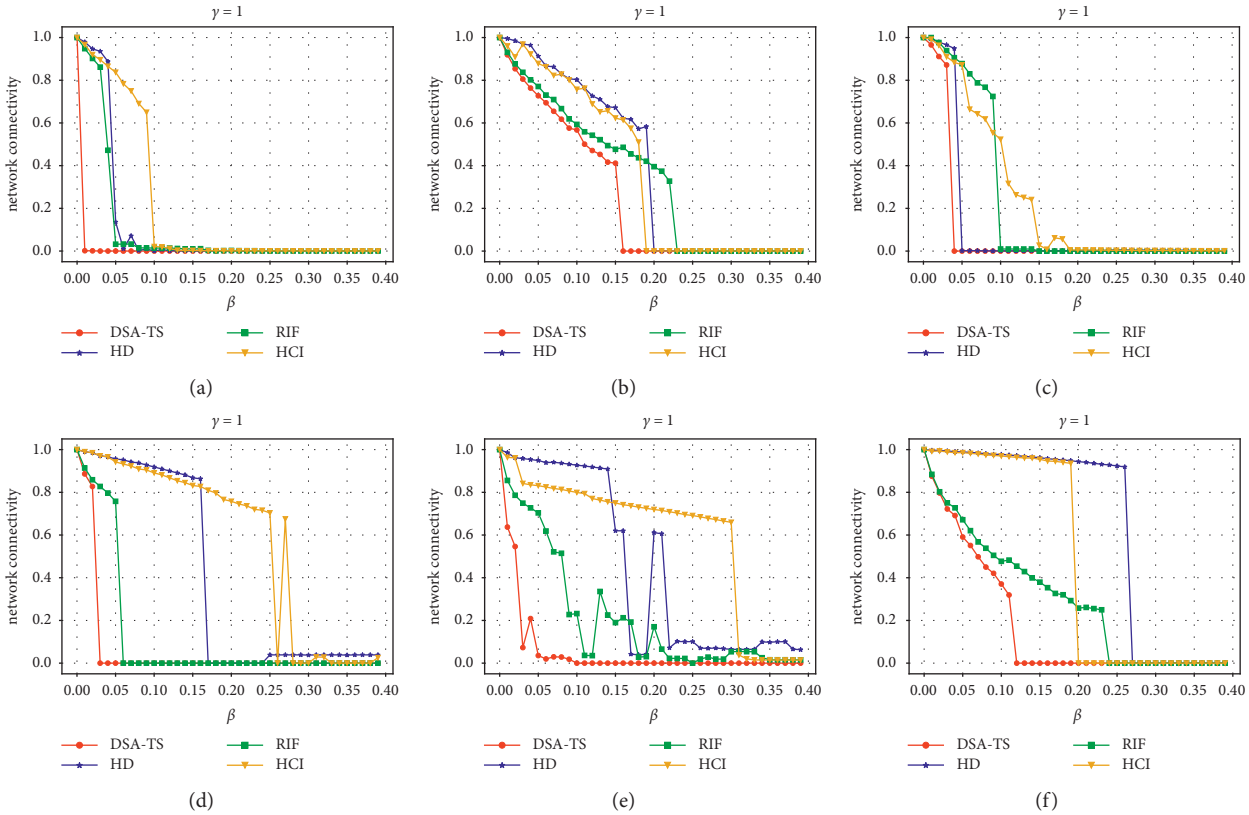


Figure 5: Effectiveness of DSA-TS applied to six real networks with different cost heterogeneities. Experimental network parameters are $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, $\lambda = 2$, and $\gamma = 1$. (a) Power grid. (b) Communication network. (c) Road network. (d) Interpersonal network. (e) Facebook. (f) Economic network.

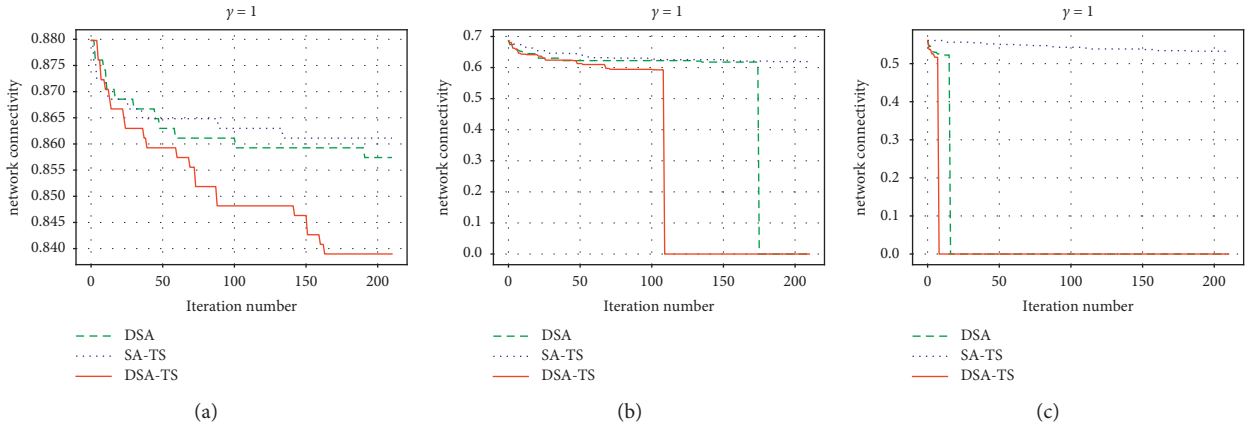


Figure 6: Convergence characteristics of algorithms under different budgets. The experimental network is a scale-free network with $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, $\lambda = 2$, and $\gamma = 1$. The total cost of the network attack is $c_{sum} = 5982$. (a) $B = 299.1$. (b) $B = 897.3$. (c) $B = 1196.4$.

vulnerability. Defense resources should be allocated reasonably according to the value of the nodes in the network.

6.2. Optimal Node Removal Characteristics under Cost Heterogeneity. This section analyzes the characteristics of the critical nodes of the network under the heterogeneity of costs. In Figure 8, the abscissa is the cost-sensitive parameter and the ordinate is the average degree of the optimal set of collapsed nodes $\widehat{d}$. Figure 8(a) shows that the average degree of the critical node set continues to decrease as the cost heterogeneity increases. At the same time, the downward trend of the average degree is an S-shaped curve, and it drops rapidly around $\gamma = 1$. The small and medium graphs in Figure 8(a) show the change in the number of selected nodes

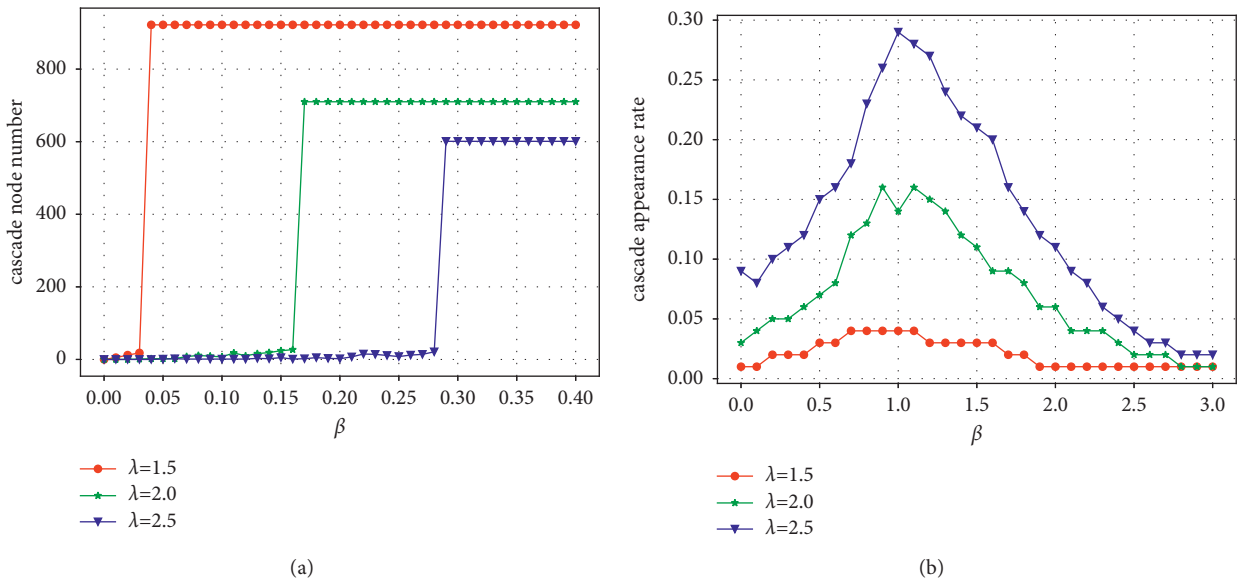


FIGURE 7: Cost heterogeneous network cascading characteristics. The experimental network is a scale-free network with $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, and $\gamma = 1$. (a) Number of network cascade nodes under different levels of node capability redundancy as the budget increases. (b) Budget of network completely cascading under different node capacity redundancy changes as the cost heterogeneity increases.

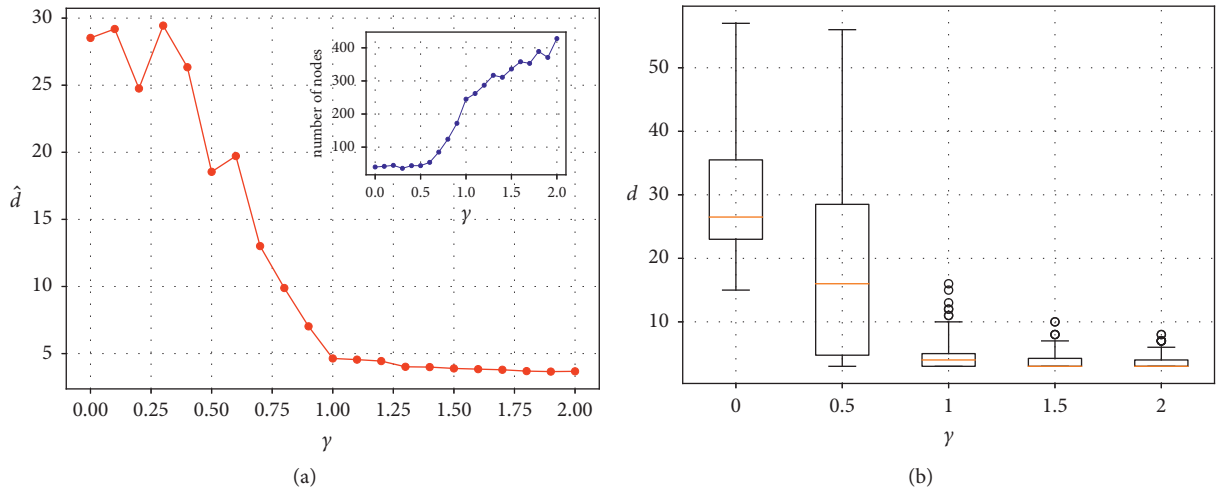
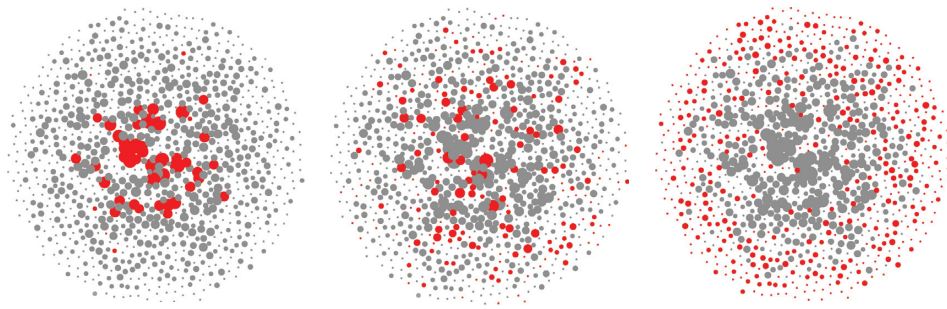


FIGURE 8: Feature box plot of node degree under completely cascaded network with different cost heterogeneities. Scale-free network with $n = 1000$ nodes and average degree $\widehat{d} = 6$, where the load parameter is $\alpha = p = 1$, the redundancy parameter is $\lambda = 2$, and the cost heterogeneity parameter is $\gamma = 0$. (a) Average degree and number of nodes (small graph) in the large-scale cascade of the network under different cost heterogeneities. (b) Box diagram of degree changes under node removal, which leads to the large-scale cascaded node degree distribution characteristics of the network under different cost heterogeneities.

with respect to the cost heterogeneity. The number of nodes rises in an S-shaped curve, with a rapid increase around $\gamma = 1$. This shows that the heterogeneity of node cost has an important influence on node selection. As the cost heterogeneity increases, more nodes with lower costs will be attacked.

To better analyze the distribution characteristics of the selected nodes under different cost heterogeneities, we present a box diagram of the removed critical nodes in Figure 8(b). From top to bottom, the box plot indicates the upper outlier, the upper edge, the upper quartile, the median, the lower quartile, and the lower edge. When $\gamma < 1$, nodes with higher degrees are all selected, and nodes with lower costs play a supplementary role. When $\gamma = 1$, the nodes with lower cost play a role, but some high-cost nodes are still selected. When $\gamma > 1$, the selected nodes have a low



(a) (b) (c)

Figure 9: Visualization of node removal under different cost heterogeneities. The scale-free network has $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, and $\lambda = 2$. The size of the nodes in the graph is related to their degree. The red node is the node that the algorithm chooses to remove, and the gray node is the node that fails due to cascading.

degree, and low-cost nodes play a leading role. Figure 9 visualizes the removed nodes under different cost heterogeneities in a scale-free network.

7. Conclusion

This paper has investigated the destruction strategy for cost heterogeneous networks using the cascading failure model. In recent years, researchers have sought to maximize the declining network performance by removing as few network nodes as possible, but the cost heterogeneity of the nodes has been ignored. This paper has proposed a heterogeneous cost model of the relationship between nodes and costs. We assumed that the cost is related to the degree of the nodes and can be adjusted by a cost-sensitive parameter γ. We found that due to the cascading characteristics of the network, different node removal orders have different effects on network performance. The DSA-TS algorithm was designed to select the sequence of nodes for removal that maximizes the declining network performance when the attacker's budget is constrained. In DSA-TS, a directional disturbance strategy improves the algorithm's convergence speed, and a tabu search and simulated annealing algorithm are merged to identify the optimal node removal order. The algorithm's effectiveness was proved through experiments on three simulated networks with different cost heterogeneities and six real networks. The convergence of different components of the algorithm was used to prove the convergence of the DSA-TS algorithm.

We conducted extensive experiments on a scale-free network and analyzed the cascading characteristics. As the cost heterogeneity increases, the budget required for complete network cascading first increases and then decreases, reaching a peak near $\gamma = 1$. From the perspective of the defender, this phenomenon shows that protection resources should be allocated according to the influence of the nodes. At the same time, we found that an increase in cost heterogeneity causes the average degree of the selected nodes to decrease along an S-shaped curve, with low-cost nodes playing a crucial role in network security. Therefore, from an attack perspective, the vulnerable nodes that threaten network security are determined not only by their influence on the network but also by their protection situation. When important nodes are overprotected, other nodes may pose a greater threat to network security.

The optimal network destruction strategy is still an open question, especially in terms of how to be adapted or extended in real or emulated environments. The current cost model is relatively simple, and there are many types of devices in the real network. Thus, in the future, we will build a more realistic cost model so that our method can be applied in reality.

Data Availability

The following are the links to the datasets used in this article: power grid, road network, interpersonal network, Facebook, and economic network (http://networkrepository.com/index.php); communication network (http://snap.stanford.edu/data/as-733.html).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (grant nos. 62002377, 62072424, 61772546, 61625205, 61632010, 61751211, 61772488, 61520106007, and NSF ECCS-1247944), Key Research Program of Frontier Sciences, CAS (grant no. QYZDY-SSW-JSC002), NSF CNS (grant no. 1526638), and National Key Research and Development Plan (grant nos. 2017YFB0801702 and 2018YFB1004704).

References

[1] L. Cui, S. Kumara, and R. Albert, "Complex networks: an engineering view," *IEEE Circuits and Systems Magazine*, vol. 10, no. 3, pp. 10–25, 2010.

[2] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.

[3] A. Azzolin, L. Dueñas-Osorio, F. Cadini, and E. Zio, "Electrical and topological drivers of the cascading failure

dynamics in power transmission networks," *Reliability Engineering & System Safety*, vol. 175, pp. 196–206, 2018.

[4] G. Stergiopoulos, E. Valvis, F. Anagnou-Misyris, N. Bozovic, and D. Gritzalis, "Interdependency analysis of junctions for congestion mitigation in transportation infrastructures," *ACM SIGMETRICS - Performance Evaluation Review*, vol. 45, no. 2, pp. 119–124, 2017.

[5] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Computers & Operations Research*, vol. 36, no. 7, pp. 2193–2200, 2009.

[6] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[7] A. E. Motter and Y. C. Lai, "Cascade-based attacks on complex networks," *Physical Review*, vol. 66, no. 6 Pt 2, Article ID 065102, 2003.

[8] M. G. Angle, S. Madnick, J. L. Kirtley, and S. Khan, "Identifying and anticipating cyberattacks that could cause physical damage to industrial control systems," *IEEE Power and Energy Technology Systems Journal*, vol. 6, no. 4, pp. 172–182, 2019.

[9] C. Chen, M. Cui, X. Fang, B. Ren, and Y. Chen, "Load altering attack-tolerant defense strategy for load frequency control system," *Applied Energy*, vol. 280, pp. 116–015, 2020.

[10] M. Cui and J. Wang, "Deeply hidden moving-target-defense for cybersecure unbalanced distribution systems considering voltage stability," *IEEE Transactions on Power Systems*, vol. 36, pp. 1961–1972, 2020.

[11] Y. Tang, G. Bu, and J. Yi, "Analysis and lessons of the blackout in Indian power grid on july 30 and 31, 2012," *Proceedings of the CSEE*, vol. 32, no. 25, pp. 167–174, 2012.

[12] K.-I. Goh, B. Kahng, and D. Kim, "Fluctuation-driven dynamics of the internet topology," *Physical Review Letters*, vol. 88, no. 10, p. 108701, 2002.

[13] M. Di Summa, A. Grosso, and M. Locatelli, "Branch and cut algorithms for detecting critical nodes in undirected graphs," *Computational Optimization and Applications*, vol. 53, pp. 649–680, 2012.

[14] Y. Shen and M. T. Thai, "Network vulnerability assessment under cascading failures," in *Proceedings of the Global Communications Conference*, Austin, Texas, December 2014.

[15] J. Seo, S. Mishra, X. Li, and M. T. Thai, "Catastrophic cascading failures in power networks," *Theoretical Computer Science*, vol. 607, no. 3, pp. 306–319, 2015.

[16] D. Huang, Q. Cao, A. Sinha et al., "New architecture for intra-domain network security issues," *Communications of the ACM*, vol. 49, no. 11, pp. 64–72, 2006.

[17] J. Zhang, S. Zhong, T. Wang, H. Chao, and J. Wang, "Blockchain-based systems and applications: a survey," *Journal of Internet Technology*, vol. 21, no. 1, pp. 1–14, 2020.

[18] J. Wang, Y. Yang, T. Wang, R. Sherratt, and J. Zhang, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[19] Z. Baig and S. Zeadally, "Cyber-security risk assessment framework for critical infrastructures," *Intelligent automation and soft computing*, vol. 25, no. 1, pp. 121–129, 2019.

[20] Y. Park, H. Choi, S. Cho, and Y.-G. Kim, "Security analysis of smart speaker: security attacks and mitigation," *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1075–1090, 2019.

[21] Q. Wei, G. Hu, C. Shen, and Y. Yin, "A fast method for shortest-path cover identification in large complex networks," *Computers, Materials & Continua*, vol. 63, no. 2, pp. 705–724, 2019.

[22] D. Zhu, Y. Sun, X. Li et al., "MINE: a method of Multi-Interaction heterogeneous information Network Embedding," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1343–1356, 2020.

[23] W. M. Eid, S. Atawneh, and M. Al-Akhras, "Framework for cybersecurity centers to mass scan networks," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1319–1334, 2020.

[24] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[25] S. Brin, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the of 7th World Wide Web Conference*, Brisbane Australia, 1998.

[26] M. Bellingeri, D. Cassi, and S. Vincenzi, "Efficiency of attack strategies on complex model and real-world networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 414, pp. 174–180, 2014.

[27] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[28] X. L. Ren, N. Gleinig, D. Helbing, and N. Antulov-Fantulin, "Generalized network dismantling," *Proceedings of the National Academy of Sciences*, vol. 116, no. 14, pp. 6554–6559, 2018.

[29] B. Liu, Z. Li, X. Chen, Y. Huang, and X. Liu, "Recognition and vulnerability analysis of key nodes in power grid based on complex network centrality," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 3, pp. 346–350, 2018.

[30] R. Aringhieri, A. Grosso, P. Hosteins, and R. Scatamacchia, "Local search metaheuristics for the critical node problem," *Networks*, vol. 67, no. 3, pp. 209–221, 2016.

[31] Y. Zhou, J.-K. Hao, Z.-H. Fu, Z. Wang, and X. Lai, "Variable population memetic search: a case study on the critical node problem," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 187–200, 2021.

[32] D. Helbing, "Globally networked risks and how to respond," *Nature*, vol. 497, no. 7447, pp. 51–59, 2013.

[33] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization," *Chaos*, vol. 17, no. 2, pp. 026103–026979, 2007.

[34] L. Zhao, K. Park, and Y. C. Lai, "Attack vulnerability of scale-free networks due to cascading breakdown," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 70, no. 3, Article ID 035101, 2004.

[35] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 200–210, 2016.

[36] L. Zhang, J. Xia, F. Cheng, J. Qiu, and X. Zhang, "Multi-objective optimization of critical node detection based on cascade model in complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 2052–2066, 2020.

[37] Y. Zhu, J. Yan, Y. Sun, and H. He, "Revealing cascading failure vulnerability in power grids using risk-graph," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3274–3284, 2014.

[38] M. Wang, Y. Xiang, and L. Wang, "Identification of critical contingencies using solution space pruning and intelligent search," *Electric Power Systems Research*, vol. 149, no. Aug, pp. 220–229, 2017.

[39] Y. Qin, X. Zhong, H. Jiang, and Y. Ye, "An environment aware epidemic spreading model and immune strategy in complex networks," *Applied Mathematics and Computation*, vol. 261, pp. 206–215, 2015.

[40] D. J. Watts, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002.

[41] Z. X. Wu, G. Peng, W. X. Wang, S. Chan, and W. Ming, "Cascading failure spreading on weighted heterogeneous networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 5, pp. 202–205, 2008.

[42] J. Wang, L. Rong, L. Zhang, and Z. Zhang, "Attack vulnerability of scale-free networks due to cascading failures," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 26, pp. 6671–6678, 2008.

[43] H. Mo and G. Sansavini, "Dynamic defense resource allocation for minimizing unsupplied demand in cyber-physical systems against uncertain attacks," *IEEE Transactions on Reliability*, vol. 66, no. 4, pp. 1–13, 2017.

[44] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[45] K. Y. Lee, *Modern Heuristic Optimization Techniques : Theory and Applications to Power Systems*, Wiley, Hoboken, New Jersey, US, 2008.

[46] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[47] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[48] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[49] P. Erdös and A. Rényi, "On random graphs I," *Publicationes Mathematicae*, vol. 4, pp. 3286–3291, 1959.

[50] R. A. Rossi and N. K. Ahmed, *NetworkRepository: An Interactive Data Repository with Multi-Scale Visual Analytics*, Eprint Arxiv, 2014, https://arxiv.org/abs/1410.3560.

[51] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the Eleventh ACM SIGKDD International Conference*, Chicago Illinois USA, August 2005.

[52] S. A. Muhammad and K. V. Laerhoven, "Discovering social circles in ego networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, pp. 73–100, 2014.

[53] P. W. Holland and S. Leinhardt, "Transitivity in structural models of small groups," *Comparative Group Studies*, vol. 2, no. 2, pp. 107–124, 1971.

[54] W. Wang, C. Qiao, Y. L. Sun, and H. He, "Risk-aware attacks and catastrophic cascading failures in U.S. power grid," in *Proceedings of the Global Communications Conference, GLOBECOM 2011*, pp. 5–9, Houston, Texas, USA, December 2011.