# An Elitism-Based Multi-Objective Evolutionary Algorithm for Min-Cost Network Disintegration

Qian Li[a,*], San-Yang Liu[a], Yiguang Bai[a], Xing-Shi He[b], Xin-She Yang[c]

[a] *School of Mathematics and Statistics, Xidian University, Xi'an, Shaanxi, P.R. China*

[b] *College of Science, Xi'an Polytechnic University, Xi'an, Shaanxi, P.R. China*

[c] *School of Science and Technology, Middlesex University, London NW4 4BT, UK*

## Abstract

Network disintegration or strengthening is a significant problem, which is widely used in infrastructure construction, social networks, infectious disease prevention and so on. But most studies assume that the cost of attacking anyone node is equal. In this paper, we investigate the robustness of complex networks under a more realistic assumption that costs are functions of degrees of nodes. A multi-objective, elitism-based, evolutionary algorithm (MOEEA) is proposed for the network disintegration problem with heterogeneous costs. By defining a new unit cost influence measure of the target attack node and combining with an elitism strategy, some combination nodes' information can be retained. Through an ingenious update mechanism, this information is passed on to the next generation to guide the population to move to more promising regions, which can improve the rate of convergence of the proposed algorithm. A series of experiments have been carried out on four benchmark networks and some model networks, the results show that our method performs better than five other state-of-the-art attack strategies. MOEEA can usually find min-cost network disintegration solutions. Simultaneously, through testing different cost functions, we find that the stronger the cost heterogeneity, the better performance of our algorithm.

Published online: *Knowledge-Based Systems*, Volume 239, 5 March 2022, Article 107944 https://doi.org/10.1016/j.knosys.2021.107944

*Keywords:* Network robustness, network disintegration, heterogeneous cost, multi-objective optimization, elitism strategy.

[*] Corresponding author

## 1. Introduction

Modern societies have many key networks such as transportation networks, power transmission networks, interconnected social networks, etc., which have huge impacts on the quality of life; thus protecting their robustness and integrity becomes crucially important [1, 2]. On the other hand, infectious disease networks, criminal networks, and terrorist organization networks also exist. Such undesirable networks should be controlled and disintegrated so as to minimize their detrimental effects on society. Therefore, studying the robustness and the weakness of networks is of practical importance [3, 4].

Network analysis suggests that the network integrity appears to be heavily related to a small number of skeleton nodes (key nodes), which seems to maintain the framework and the performance of network[5, 6]. In essence, the problem of network disintegration is equivalent to finding the optimal (minimum) set of these key nodes that may strongly influence the structural integrity of the network [7, 8]. Consequently, we can enhance the robustness of the network by protecting these key nodes to ensure the smooth running of the transportation and logistics networks. On the other hand, the removal of some key nodes can maximally fragment some undesirable networks [9], which potentially provides key insight into the ways of controlling diseases and isolating certain network nodes [10].

Network robustness has received much attention in the past. The network disintegration is a very challenging problem; in fact, it is a non-deterministic polynomial-time (NP) hard. Thus, it is unlikely to have any efficient methods, most network attack strategies are still based on heuristic ranking to identify influencing nodes [11]. Subsequently, various methods were proposed for tackling this problem, which were usually approximation algorithms based on different theories or assumptions, including the optimal percolation theory [12], module-based attacks [13], the collective influence (CI) algorithm [14], or the Min-Sum algorithm [7].

These methods based on the centrality of nodes treat the optimal dismantling set as a collection of “well-performing” nodes. However, This problem is essentially a collective problem [7]. Moreover, these methods are not global optimization methods and cannot guarantee the optimality of the solution [15]. In addition, studies have shown that many weakly connected nodes that may be critical to the network structure were previously ignored[16]. Subsequently, metaheuristic algorithms were used to solve the network dismantling problem [17, 18]. These black-box types of global optimization algorithms were considered to be more suitable for this problem.

However, an important issue is that most of the existing studies on network robustness have an implicit assumption: the removal cost of any node in the network is equal, regardless of its centrality or importance. This assumption is not valid for many real-world networks. For example, the removal of a hub node can be more costly than the other nodes. Also, it is more difficult and costly to arrest the leader of a terrorist organization than its ordinary members. In addition, when an infectious disease breaks out, it is more effective to isolate super-spreaders without symptoms than the simple isolation of ordinary patients. In the case of the limited resources and environment, it is more realistic to take account of the heterogeneous costs in the strategy or model. Recent attempts in this respect have been carried out, the cost of protecting or attacking a node was defined as a function of its degree [19, 15], but there are still some shortcomings, which we will explain in detail later.

A crucial issue is that existing methods do not really incorporate the cost factor into the optimization process. When the cost function of the nodes changes, the solutions obtained by these methods will not change accordingly. So it is difficult to find a low-cost and efficient set of key nodes. It can be expected that attacking a higher-degree node in a social system can usually incur a higher cost than the same operation on a lower-degree node. A reasonable attack strategy should find the optimal node or nodes with lower costs, and the removal may cause more damage to the network. However, the key issue is that the lower costs and higher damage seem to be conflicting. Existing methods struggle to deal with this type of problem.

In fact, multi-objective metaheuristic algorithms are very powerful for solving these challenging problems [20] and good at dealing with several conflicting objectives [21, 22]. In contrast with single objective optimization, there are multiple optimal solutions in multi-objective optimization, which form the so-called Pareto Front. When the decision-maker needs to consider some practical constraints, such as the execution budget and conditions, Pareto optimal solutions can provide different choices or options for the decision-maker.

Motivated by the above challenges, we now take a multi-objective metaheuristic approach to deal with such a problem. This paper proposes an elitism-based multi-objective evolutionary algorithm for network disintegration problem with heterogeneous costs. In essence, the problem is treated as a bi-objective problem: one objective is the cost of protecting or removing/attacking nodes; the other is the extent to which the network is disintegrated after removing the nodes. We assume that the removal cost of a node can be either an exponential function or a power function of the node's degree. Our proposed approach intends to find a set of nodes with the lowest removal costs for network collapse and simultaneously try to provide more suitable choices for decision-makers with limited resources. Thus, the main contributions of this work can be summarized as follows:

1. The network disintegration problem with heterogeneous costs is formulated for the first time as a bi-objective problem by incorporating the cost as one of the objective functions. A multi-objective, elitism-based, evolutionary algorithm (MOEEA) is proposed to solve this problem, which can find a set of better solutions efficiently.
2. A new unit cost importance measure is defined, which combines attack cost and node importance to provide a measure for the comparison and selection of nodes in multi-objective problems.
3. The combination influence of nodes has been considered to the proposed algorithm. A reservation mechanism combining a unit cost influence measure is proposed for elite individuals. The reserved key node combination information will participate in each offspring's individual generation process and guide the population to move to more promising regions.
4. The parameter $N_s$ is used to convert between local search and global search to achieve a balance between exploitation and exploration.

Therefore, this paper is organized as follows. Section 2 discusses the recent developments, whereas Section 3 introduces some background concerning the estimation of measures and the optimization formulation for complex network disintegration with heterogeneous costs. Section 4 focuses on the proposed elitism-based multi-objective evolutionary algorithm, including the preprocessing, elitism strategy, update mechanism, and the realization of non-dominant solutions and selection mechanisms. Then, Section 5 presents a series of experiments on real networks and widely used network models, followed by a summary of the comparison with five other methods. Algorithm complexity analysis and some discussions are carried out in Section 6. Finally, Section 7 draws some conclusions and discusses briefly the relevant topics for further research.

## 2. Recent Developments

Complex networks exist in many areas, such as biological/ecological networks, disease transmission networks, logistics networks and others. Many studies have focused on network robustness in recent years, and different approaches and methods have tried to tackle the network dismantling problem, though the results are mixed.

The importance of a node on a network is highly related to its ability to influence the behavior of its neighbors. Therefore, an effective method is to directly calculate the number of neighbors (i.e., the degree of nodes). Degree centrality is widely used because of its simplicity and low computational complexity, and it often shows good performance. For example, in the study of network vulnerability, compared with the betweenness, closeness and eigenvector, attacks according to degree centrality can effectively destroy scale-free networks and exponential networks [23]. In addition, when the propagation rate is very low, the degree centrality can reflect the diffusion effect of nodes better than other centrality [24, 25].

The most classic disruptive method is called the High Degree First (HDF) [26], which sorts the nodes in the network according to their degrees, and then the nodes with the highest degree are attacked first. The advantage of HDF is its low computational complexity. High Degree Adaptive (HDA) method [27] is an adaptive version of HDF. HDA recalculates the degrees of all remaining nodes after deleting a node, then carries out sorting again. However, the degree of a node is a local characteristic, not a global feature relative to the entire network [28, 29]. In other words, only the nearest neighbor information is considered in these methods. Therefore, the performance of these methods is not as efficient as expected.

By the energy minimization of a multi-body system, the collective influence (CI) essentially maps a random network onto an optimal percolation to find out the minimal set of influencers [12]. This CI indicator of nodes is calculated from the degree of neighborhoods with a radius $\ell$. As the adaptive approach, the CI value of all nodes in the remaining network is calculated at each iteration, and then the node with the highest CI value will be deleted. The advantage of CI algorithm is that it can find some weak connection key nodes.

Betweenness is another widely used centrality measure. The betweenness of a node refers to the percentage of the shortest path between any two nodes passing through the node [30]. Generally speaking, a node with high betweenness is equivalent to a bridge connecting two communities, which usually plays an important role in the network. The High Betweenness First (HBF) [31] strategy is to sort the nodes according to their betweenness, and then remove them from large to small. However, as more nodes are attacked or deleted, the network structure will change, resulting in the betweenness distribution may be very different from the initial network. Therefore, a

High Betweenness Adaptive (HBA) method [32, 33] is proposed. HBA is a self-adaptive method. Before deleting each node, it recalculates the betweenness values of nodes and resorts them. Compared with the method based on degree centrality, the method based on betweenness has higher computational complexity, but it has some global properties.

All these strategies seem to be heuristic ranking [34, 2], which was based on graph theory [9, 34, 35]. A serious issue is that these methods treat the optimal dismantling set as a collection of nodes rather than a set of most effective combination of nodes [7].

Subsequently, a two-step method is proposed by Anggraini et al. [36]: the given network was divided into several isolated communities, and then the critical nodes in each community are eliminated, respectively. However, community detection is still a difficult problem, and the key nodes in the community may not be the same as the globally influential nodes. Braunstein et al. [7] presented a three-stage minimum sum algorithm, and its three stages are: a) min-sum message-passing algorithm was applied to dismantling, b) a large component was broken into small ones by a greedy procedure, and c) some nodes were reinserted to improve the algorithm efficiency.

Some preliminary studies used metaheuristic algorithms in network robustness or integrity. For example, Deng et al. suggested an optimized attack strategy model and described the tabu search to dismantle networks [17]. However, this algorithm is a greedy approach, which may take a considerable time cost to find an acceptable solution. Li et al. [18] introduces a metaheuristic algorithm: a probabilistic algorithm, based on neighborhood information. In their paper, a novel importance measure (IM) based on centrality was defined, and the combination of nodes was considered in the algorithm iteration process.

Although some network dismantling strategies showed promising results, these algorithms have been based on an assumption: the cost of removing any node is the same. This assumption is not suitable for any realistic networks. For example, the cost of maintaining critical infrastructure will be much higher than the cost of some conventional auxiliary facilities. In fact, the same action on high degree nodes can usually incur a higher cost in the system [15]. Therefore, a more sensible approach is to assume that the removal cost is heterogeneous. The good news is that some researchers started to investigate this for the problem of network disintegration with heterogeneous costs.

Deng et al. [37] proposed an optimal strategy based on a limited cost model, where the resource-limited condition is taken as the termination condition, and the attack nodes set is formed by gradually increasing the number of nodes with a high probability. However, in essence, the so-called unequal probability sampling method is the same as the 'HDA removal' method [27], which requires recalculating the selection probability in the remaining nodes at every iteration. If $\alpha \geq 0$, the node with the highest degree will be attacked, whereas, if $\alpha < 0$, the node with the lowest degree will be selected. This approach is essentially a node combination satisfaction condition, not a global optimization problem.

Patron et al. [19] defined a parameter $z(k)$, which indicates the contribution relationship of the unit cost of removing all $k$ degree nodes to the network destruction. Therefore, the value of $z(k)$ for each degree $k$ is calculated first, and then the attack is carried out in descending order. It is worth noting that if $z(k)$ is selected, all nodes with degree $k$ will be removed until the network is completely disintegrated. The node-set of the last selected degree may be partially attacked. Their experimental results showed that the elimination priorities according to the value of $z(k)$, could be more effective. This method partially considered the cost factor as part of the problem. However, this method is still not a global optimization approach.

Deng et al. [38] presented a network disintegration model under a cost constraint with three optimal attack strategies: the hub strategy (eliminating nodes in descending order of degrees), the leaf strategy (eliminating nodes in ascending order of degrees), and the average strategy (preferentially deleting nodes close to the average degree). The effects of the three strategies were discussed when the attack cost was heterogeneous. Their experiment showed that the average strategy had an excellent disintegration effect. This method can be considered as a simple variation of the heuristic sorting method.

Ren et al. [15] presented an approach that the disintegration cost of a node is proportional to or equivalent to the degree of the node. When considering the attack cost, some of the existing algorithms for node/edge removals became inefficient. In their paper, they introduced a modified edge removal method, called the hierarchical power iterative normalized cut (HPI N-cut). Their experimental results showed that their algorithm was superior to almost all the node deletion and link deletion algorithms at the same cost. However, this approach is still a single-objective approach.

In contrast, our new approach in this paper is a multi-objective approach to solve the network disintegration problem with heterogeneous costs. An elitism-based multi-objective evolutionary algorithm is proposed to identify the set of key nodes for both strengthening networks and potentially destroying networks with the minimum cost.

## 3. Background

### *3.1. Basic Definitions*

An undirected network can be expressed as a graph $G = (V, E)$ with the number of $N = |V|$ nodes and the number of $M = |E|$ edges. Its adjacency matrix can be represented as an $N \times N$ matrix $A = (a_{ij})_{N\times N}$. In addition, it is required that $a_{ij} = a_{ji} = 1$ for any two adjacent nodes $\mathrm{v}_i$ and $\mathrm{v}_j$.

When some nodes are removed from the network, the structure will be destroyed, and the whole connectivity will be lost. The remaining network may become several disconnected branches (the interior of each branch is connected). The branch (sub-clusters) with the largest number of nodes is called the largest connected branch. For a network with a probability distribution of $P(k)$ for the degrees of nodes, if it contains the largest connected branch, the nodes connected by any edge in this branch must connect to at least another edge. In other words, the average degree is at least 2 to allow the existence of the largest connected branch in the network [6]. For example, node $i$ and node $j$ are connected with each other such that the average degree is

$$\mathbb{E}[k] = \langle k_i \,|i \leftrightarrow j\,\rangle = \sum_{k_i} k_i P(k_i \,|i \leftrightarrow j\,) = 2, \tag{1}$$

where $k_i$ represents the degree of node $i$, $P(k_i \;|i \leftrightarrow j)$ corresponds to the probability that node $i$ is connected to node $j$ and the degree of node $i$ is $k_i$. According to the Bayesian rule, $P(k_i \;|i \leftrightarrow j)$ is equal to

$$P(k_i \;|i \leftrightarrow j)/P(i \leftrightarrow j) = P(i \leftrightarrow j|\; k_i)P(k_i)/P(i \leftrightarrow j). \tag{2}$$

For a random network with $N$ nodes (ignoring loops), the following formulas hold: $P(i \leftrightarrow j) = \langle k\rangle/(N-1)$, and $P(i \leftrightarrow j\,|k_i\,) = k_i/(N-1)$. Thus, Eq. (1) becomes

$$\kappa \equiv \frac{\mathbb{E}[k^2]}{\mathbb{E}[k]} \equiv \frac{<k^2>}{<k>} = 2. \tag{3}$$

This is regarded as the critical disintegration threshold. When $\kappa > 2$, the largest connected branch exists. On the other hand, $\kappa < 2$ means that the graph only contains small connected branches, and thus the graph can be regards as completely disintegrated [29].

The largest connected branch is an important indicator. If the largest connected branch in the remaining network contains the majority of nodes, it means that the network has not been greatly damaged. On the contrary, if the largest connected branch contains only a few nodes, it means that the network has lost most of its connectivity and suffered great damage.

Before the network is completely disintegrated, we use the classical measure $S(Q)$ in this paper to estimate the extent of the network damage. Here, $S(Q)$ is equal to the percentage of nodes in the largest branch after deleting $Q$ nodes among the total number of nodes in the original network. Note that if the network is considered to have been completely disintegrated ($\kappa < 2$), the value of $S(Q)$ is equal to 0. In essence, the value of $Q$ varies from 0 (no nodes were attacked) to $N$ (all removal), and $Q$ is the attack intensity. When the network completely disintegrates, it is considered that the network does not contain the largest connected branch, so $S(Q)$ is 0. Furthermore, to evaluate the effectiveness of an attack strategy, a critical attack strength $Q_c$ is often calculated, which can be considered as the minimum attack strength needed to completely disintegrate the network [39, 40].

### *3.2. Optimization Formulation*

It is unrealistic to assume that the attack cost of all nodes is equivalent. As the centrality (importance) of the node increases, the cost of destroying and strengthening the node shall also increase. Ren et al. [15] assumed that the cost of a removing action was proportional to the node's degree. Such cost definition is reasonable, but is a relatively simplest approach. To be realistic and practical, the removal cost of a node can be as a function of its degree. For example, Patron et al. [19] defined the cost function $c(k_i)$ as:

(i) an exponential function:

$$c(k_i) = e^{\beta k_i}, \quad \beta > 0, \tag{4}$$

(ii) a power law:

$$c(k_i) = (k_i)^{\gamma}, \quad \gamma > 0. \tag{5}$$

where $k_i$ is the degree of the node $i$. For the exponential function, the costs of nodes have strong heterogeneity. In this paper, we will use both the exponential and power-law functions.

The additional consideration of costs for network disintegration leads to a multi-objective problem for designing network attack strategies. One objective is to maximize the extent of the damage, and the other objective is to minimize the cost of removing target attack nodes. In fact, these two goals are conflicting. As the centrality of a node increases, the cost will usually increase. Removing nodes with high centrality or increasing the number of nodes will destroy the network to a greater extent, but increased costs usually accompany this. This is the nature of multi-objective optimization, the change of one objective will affect the other (or other objectives), and thus Pareto optimal sets should be sought. When an attacker needs to consider some restrictions, such as limited budget and time constraints, Pareto optimal solutions provide different attack strategies or options, based on existing resources, in the non-dominated sense.

It is noteworthy that such conflict is not contradictory; instead, they are non-dominated solutions or options. The first objective is to minimize the proportion of the largest connected branch in the network after removing nodes. Hence, both objectives are minimization, not one maximization and one minimization.

The current state of the network is represented by an $N$-length binary string $(x_1, x_2, \cdots, x_N), x_i \in \{0, 1\}$, where $x_i = 1$ indicates the status of node $i$ is existence, whereas $x_i = 0$ indicates that the node $i$ has been removed or attacked. In this paper, we assume that the attack cost of a node depends on its degree. If the degree of a node $i$ is $k_i$, the cost function of removing this node is $c(k_i)$, as given in Eq. (4) or Eq. (5). Hence, the total cost for all $N$ nodes is

$$C = \sum_{i=1}^{N} (c(k_i) * x_i). \tag{6}$$

For multi-objective optimization, the objectives should be in the same order so as to produce well-scaled or balanced Pareto Fronts. In our bi-objective problem, the first objective is a percentage, so we can add a weighting factor $W$ to balance these two objectives. Therefore, the problem can be formulated as a bi-objective binary integer programming problem subject to relevant constraints. The goal is to find a set of binary strings satisfying the conditions, such that both $S(Q)$ and $C$ are minimized:

$$\text{Minimize} \qquad F = \{f_1 = W * S(Q), \quad f_2 = C\} \tag{7}$$

subject to

$$\sum_{i=1}^{N} x_i = N - Q, \quad x_i \in \{0, 1\}, \quad \forall i. \tag{8}$$

where $Q$ is the attack strength, and $S(Q)$ is the fraction of nodes in the largest connected cluster after removing $Q$ nodes. $W$ is the weight factor, we use $W = 100$ in this paper.

This is essentially a bi-objective and binary integer programming problem. Since $N$ can be very large, such problems can be NP-hard. Though there are no efficient algorithms for solving such issues in general, approximation methods and metaheuristic methods seem to be good alternatives. Recently, some metaheuristic algorithms have been successfully applied to deal with these problems with promising results. For example, a new firefly multi-objective optimization algorithm based on chaos mechanism is used to detect network communities [41]. A multi-objective evolutionary algorithm based on structure and attribute similarity is used to solve the community detection problem in attribute networks [42]. Zhang et al. proposed a multi-objective evolutionary algorithm based on network simplification to solve the problem of community detection in large-scale networks [43]. However, there is no method yet to solve the problem of network disintegration, and there is no effective way to deal with the cost factor.

## 4. Elitism-based multi-objective evolutionary algorithm

Finding the best combination of attack nodes with the minimum total removal cost in the network is computationally expensive. Such a challenging problem becomes even more complicated when dealing with multi-objective optimization. In this paper, we propose a multi-objective, elitism-based, evolutionary algorithm (MOEEA), which uses an elitism mechanism with the approach of reservation information. The initialization, elitism strategy, update mechanism, and other details will be explained in detail below.

### *4.1. Initialization*

Initialization is a crucial step for metaheuristic algorithms [44]. In fact, proper initialization can greatly improve the performance of the algorithm. On the one hand, we hope to obtain some guidance through some prior knowledge of the problem, on the other hand, we hope that the population has good diversity. Hence, the initial population (of size $N_P$) is generated in two categories for the network dismantling problem.

In the first part, some prior knowledge and the information can be used to guide the generation of solutions. Considering some centrality measures, such as: the degree [28, 29], betweenness centrality [34], and the collective influence [12] are widely used to identify the importance nodes in the network [45]. In addition, this information is easily available. Therefore, some individuals (of size three in this paper) are generated according to the degree, betweenness, and collective influence of nodes. Specifically, by ranking the nodal degrees in descending order, the first $Q$ nodes are selected as the initial attack nodes. The bits of the binary string corresponding to these selected attack nodes are 0s, while the remaining bits are 1s. Similarly, the other two types of individuals are generated by ranking betweenness and collective influence, respectively.

For the second part of the population, we perform the following operations: $Q$ nodes are randomly selected from the network as attack nodes, where the bits of binary strings corresponding to these nodes are 0s, and other bits are 1s. Hence, $(N_P - 3)$ initial solutions can be generated, and each of them is an $N$-length binary string $X = \{x_i | i = 1, 2, \cdots, N\}$. In this way, the diversity of population can be improved.

These two ways together produce $N_P$ binary strings (individuals). All $N_P$ individuals form the initial population, which can be considered as evolutionary parents.

### *4.2. Definitions of unit cost important measure*

Some classic disassembly strategies focus on those nodes with high centrality, but ignore the interaction between nodes in the network. As a result, some weakly-connected vulnerable nodes are often ignored. However, studies have shown that some low-centrality nodes may surprisingly be crucial to the network integrity [12]. The loss of certain information leads to the effectiveness of attack strategies based on these measures is not as high as expected. A new measure is highly needed to evaluate the importance of nodes.

Hence, Li et al. [18] proposed a novel centrality-based measure for network disintegration problem: the importance measure (IM), which can be understood as how many nodes are expected to be separated from the largest connected branch by attacking the target node. IM can practically better evaluate the importance of the node to the network collapse. Not only the node's neighbor information, but also the two-hop node information is considered. The definition is briefly described as follows:

**Definition 1.** *Suppose that the target attack node $j, j \in (1, 2, \cdots Q)$, has $T$ directly connected neighbors and the t-th neighbor node is expressed as $j_t, j_t \in (1, 2, \cdots, T)$. The importance measure (IM) of node $j$ is defined as the sum of the product of the degree of each neighbor node and the corresponding contribution ratio [18]. The contribution ratio $P_{j_t}$ of node $j_t$ can be calculated by the reciprocal of the number of all target attack nodes that are linked directly to this neighbor node under consideration [18]. It is worth pointing out that all neighbor nodes discussed here are not included in the largest* *connected branch, after the target node is removed from the network. That is*

$$IM_j = \sum_{j_t=1}^{T} P_{j_t} k_{j_t}, \tag{9}$$

$$P_{j_t} = \frac{1}{|v|}, \quad v = \{k \mid a_{j_t,k} = 1, \ \ k \in \Phi\}, \tag{10}$$

where $k_{j_t}$ represents the degree of the node $j_t$. Some neighbor nodes may be connected to more than one target attack node at the same time. Here, $v$ represents the set of attack nodes that are directly connected to the node $j_t$, and $|\cdot|$ indicates the total number of elements in the whole set. The set of target attack nodes is represented as $\Phi$.

When the cost is considered, we can define the unit cost importance measure for network disintegration:

**Definition 2.** *The unit cost importance measure (UIM) is defined as the quotient of the important measure (IM) to the attack cost of the corresponding node. That is*

$$UIM_j = \frac{IM_j}{c_j}, \tag{11}$$

where $IM_j$ and $c_j$ correspond to the importance measure (IM) and the cost of $j$-th target attack node, respectively.

The unit cost important measure can be calculated by Algorithm 1. With the above definition of $UIM$, it may be possible to select those more destructive nodes to the network, but with a lower cost or the same cost. This allows to disrupt a network and to reduce the attack cost. We will elaborate on this point later when discussing the elitism strategy.

### *4.3. Elitism strategy*

Many existing methods based on degree centrality focus on a single node's properties, leading to the loss of some structural information. However, the role of some weakly connected nodes cannot be ignored, and the attack efficiency of combined nodes may be better than that of some high-degree nodes. In fact, the network as a complex system has cascading effects (interactions) between nodes. As an illustrative example, shown in Fig. 1, the two nodes with the highest degree are Node 1 and Node 14. After attacking these two nodes, there are 8 nodes (including Node 6, Node 7, Node 8, Node 9, Node 10, Node 11, Node 12, Node 13) in the largest connected

**Algorithm 1** Calculate the unit cost important measure.

```
**Input**: the adjacency matrix $A$, the attack intensity $Q$, the binary string $X^{now}$, the node $i$ with degree $k_i$, and the number of nodes $N$.
$\mathbf{d} \leftarrow$ degree of nodes
$\Phi \leftarrow$ the set of 0 bits in $X^{now}$ // target attack nodes index set
$A_{temp} \leftarrow A$
$A_{temp}(\Phi, :) \leftarrow \mathbf{0}_{Q\times N}$ ; $A_{temp}(:, \Phi) \leftarrow \mathbf{0}_{N\times Q}$ //the adjacency matrix corresponding to $X^{now}$
Get the node set $\Im$ of the largest connected branch from $A_{temp}$
$\tilde{A} \leftarrow A$
$\tilde{A}(\Im, :) \leftarrow \mathbf{0}$; $\tilde{A}(:, \Im) \leftarrow \mathbf{0}$;// delete the rows and columns (corresponding to the nodes of set $\Im$) in the adjacency matrix.
for $i = 1, 2, \cdots, Q$ do
    $\Omega(\Phi(i)) \leftarrow$ the neighbor nodes of $i$-th element of $\Phi$ // based on $\tilde{A}$
end for
for $j = 1, 2, \cdots, Q$ do
    if $\Omega(\Phi(j)) == \emptyset$ then
        $I(\Phi(j)) = 0$
    else
        $IM(\Phi(j))$ // calculate by Eq.(9)
        $UIM(\Phi(j)) = \frac{IM(\Phi(j))}{c(\Phi(j))}$
    end if
end for
return $UIM$ as a vector.
```

branch. Obviously, the attack effect is far worse than that of the combination of Node 1 and Node 9, although the degree of Node 9 is not high. The attack efficiency may be better by adding some weak-connected key nodes [18] and the combination of nodes should be taken seriously.

In our MOEEA, elite individuals are used to store the better ‘combination information of nodes’ in each iteration, and guide each ‘parent individual’ to generate new offsprings.

The fitness value of each individual $X_k(k = 1, 2, \cdots, N_P)$ in the population is $F(k) = (f_1(k), f_2(k))$, where $f_1(k)$ represents the extent of network damage (the proportion of the remaining maximum connected branch), and $f_2(k)$ represents the attack costs. For the initial population, we assume that the first individual is an elite individual. By calculating each individual’s fitness value, if the two objective function values of an individual are both

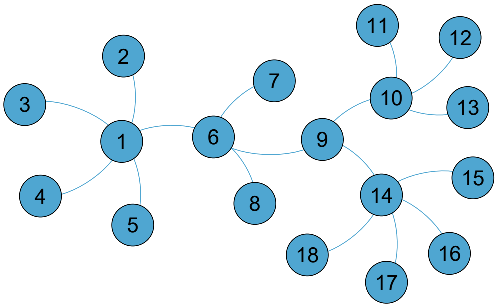


Figure 1: Examples of node combination: the attack effect of two nodes with higher degree (Node 1 and Node 14) is worse than another combination: Nodes 1 and 9.

less than or equal to the elite individual, a new elite individual is generated to replace the previous one.

The effectiveness of elite individuals may be due to specific combinations of some well-performing nodes. In order to make these right 'genes' have a chance to pass on to the next generation, the elitism strategy in this paper will be used in evolutionary process of the algorithm. Together with the reservation mechanism, this approach will guide the population to move towards more promising regions that may contain the optimality.

More specifically, according to the unit cost importance measure ($UIM$), we can estimate the 'performance' of those target attack nodes in the elite individual. At each iteration, the target attack nodes of the elite are sorted in an ascending order according to the $UIM$. Then, a certain number of attack nodes (in percentage) are stored to guide the generation of offsprings. That is, $\lfloor \alpha Q \rfloor$ are stored, where $\alpha \in (0, 1)$, and $\lfloor \cdot \rfloor$ means round toward negative infinity. These reserved nodes will not be changed in subsequent update mechanism. We use the sparsity of the adjacent matrix and only update the part of the unreserved nodes. As a time-saving strategy, this elitism not only preserves the combination of nodes as useful information, but also improves the convergence of the algorithm. How to use combination information to guide the population search process will be described in detail later in the update mechanism.

For each offspring generated in the next generation, its fitness function value is calculated. If $f_1(k) \leq f_1(elite)$ & $f_2(k) \leq f_2(elite)$, then individual $k$ becomes the new elite individual. In this way, the elitism strategy allows the better individuals in the population to be stored and used for the algorithm update.

### *4.4. Update mechanism*

A suitable update mechanism can produce better offsprings from parents. Each time an individual is selected from the 'parent population', an offspring individual is generated through the update mechanism until all the $N_P$ parents generate the corresponding offsprings (the population size in the next iteration is still $N_P$). The combination of information stored by the elite individual will participate in the generation process of each offspring individual.

In a binary string representation, the bit with status 1 represents a node that is not attacked, and the set of these nodes is represented by $\Psi$. Nodes with status 0s represent the target attack nodes, which are represented by the set $\Phi$. The set of nodes stored by the elitism strategy is denoted by $\Lambda$. For a parent individual, we randomly select $N_s$ bits to be updated. Obviously, it is required that the number $N_s$ is less than $\min\{|\Psi|, |\Phi|\}$, where $|\cdot|$ is the number of the elements of the set. Note that $N_s$ is not a fixed value, but needs to be regenerated for each parent individual. A small value of $N_s$ means that the 'parent individual' updates only a small number of 'genes' and the algorithm undergoes a local search. If the randomly generated $N_s$ is large, the algorithm makes a big jump and avoids getting trapped in local optimums. $N_s$ is actually used to convert between local search and global search to achieve a balance between exploitation and exploration.

Since there are three possible relationships between sets $\Lambda$ and $\Phi$, the update mechanism is divided into three types:

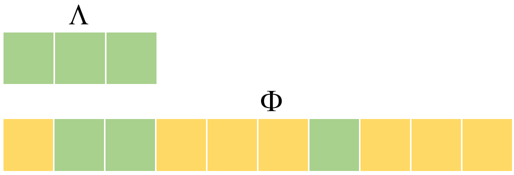


Figure 2: Schematic representation of the relationship between sets $\Lambda$ and $\Phi$ (Type A: $\Lambda \in \Phi$).

(1) Type A: $\Lambda \in \Phi$

In this case, the nodes reserved in set $\Lambda$, according to elite individuals, are included in the targeted attack node-set, as shown in Fig. 2. This means that the optimal node combination has been considered. Therefore, $N_s$ nodes are randomly selected in $\Phi$, and their states are changed from 0 (removed) to 1 (present), and then another $N_s$ nodes are randomly chosen from $\Psi$ and their states are changed from 1 to 0.

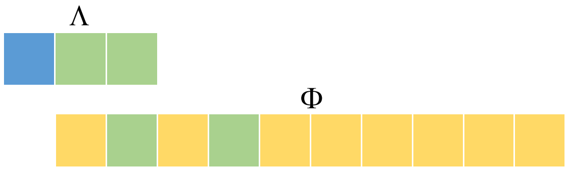


Figure 3: Schematic representations of the relationship between sets Λ and Φ (Type B: $\Lambda \cap \Phi \neq \emptyset$ & $\Lambda \notin \Phi$).

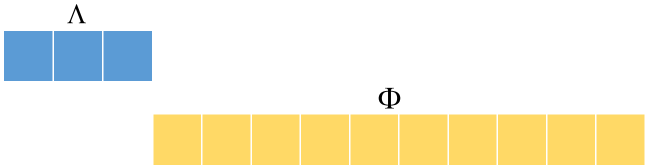


Figure 4: The third type of the relationship between sets Λ and Φ (Type C: $\Lambda \cap \Phi = \emptyset$).

(2) Type B: $\Lambda \cap \Phi \neq \emptyset$ & $\Lambda \notin \Phi$

As illustrated in Fig. 3, there are two scenarios. To be more specific, if $N_s < |\Lambda - \Lambda \cap \Phi|$, $N_s$ nodes (their states are 1s) are randomly selected in set $\Lambda - \Lambda \cap \Phi$ and their states are changed to 0. At the same time, $N_s$ nodes are randomly selected from the set $\Phi - \Lambda \cap \Phi$. In fact, the states of these corresponding bits in the binary string are 0s, and such states will be changed to 1.

On the other hand, if $N_s > |\Lambda - \Lambda \cap \Phi|$, all states of the nodes in set $\Lambda - \Lambda \cap \Phi$ are turning into 0. Additionally, $N_s - |\Lambda - \Lambda \cap \Phi|$ nodes are selected randomly from the set $\Psi - \Lambda \cap \Psi$ and their corresponding bits become 0. In this way, $N_s$ target attack nodes are newly added, So the $N_s$ nodes need to be randomly selected and removed from the attack node set $\Phi$; that is, their status changes to 1.

(3) Type C: $\Lambda \cap \Phi = \emptyset$

In this relationship, as shown in Fig. 4, there are also two scenarios. If $N_s < |\Lambda|$, $N_s$ nodes are randomly selected in set Λ to change the states of their corresponding bits from 1 to 0, then $N_s$ nodes in set Φ are chosen randomly and the states of their corresponding bits are changed from 0 to 1.

On the other hand, if $N_s > |\Lambda|$, the states of nodes in Λ should be changed from 1 to 0 at first, then $N_s - |\Lambda|$ nodes are chosen from set $\Psi - \Lambda$ and the states of these nodes are changed from 1 to 0. In addition, the $N_s$ nodes in set $\Phi$ are chosen randomly and their states of their corresponding bits are changed from 0 to 1.

### *4.5. Non-dominated solutions and selection mechanism*

For multi-objective optimization, Pareto optimal sets are obtained, which can provide options and trade-offs between the objectives. Briefly speaking, an $n$-dimensional vector (or a solution) $\mathbf{u} = (u_1, u_2, \cdots u_n)$ is said to dominate another vector or solution $\mathbf{v} = (v_1, v_2, \cdots v_n)$, if and only if $u_k \leq v_k$ for $\forall k \in \{1, 2, \cdots, n\}$ and the inequality becomes strictly inequality for at least one $k$ among all components $(1, 2, ..., n)$. Therefore, a solution $X_*$ is called a non-dominated solution if no solution $X$ can be found such that $F(X)$ dominates $F(X_*)$ [46].

The fast nondominated sorting (FNS) is proposed by Deb et al. [47], which is widely used multi-objective optimization. By the way, although MOEA/D [48] is also a widely used framework for multi-objective optimization, it is not good at solving combinatorial optimization problems (the premise of decomposition is that any solution in the neighborhood of the current solution is still the solution of the problem). Therefore, we use the FNS to sort and rank solutions in each iteration in this paper. Each individual is assigned a rank that reflects its nondomination level (the best level is 1 with no other solution dominating it, the next best level is 2, and so on).

To be more specific, at the beginning of the algorithm, the initial population can be regarded as the parent $P_0$. By calculating each individual's objective fitness value, the elite individual is selected by the elitism strategy. The parent population generates offspring population through the update mechanism (the offsprings $S_0$ composed of $N_P$ individuals). Then, a combined population $R_0 = P_0 \cup S_0$ is formed, with a size of $2N_P$. This joint population is then sorted by the NFS, leading to a new population of $N_P$ individuals for the next generation.

### *4.6. The main steps of MOEEA*

The main steps of the proposed MOEEA algorithm can be divided into seven steps, which can be summarized as follows:

- Step 1: Initialize the algorithmic parameters and the population $P_0$: the total number of network nodes $N$, the attack strength $Q$, the population size $N_P$. Calculate the fitness values of $P_0$.

- Step 2: The termination criterion: When the number of iterations reaches the preset value $T$ =iteration$_{\max}$, or the network meets the complete disintegration condition $\kappa \leq 2$, the iteration stops.

- Step 3: According to the elitism strategy, find the elite individual and calculate the unit cost important measure ($UIM$) of its target attack nodes.

- Step 4: Sort $UIM$ in a descending order, and select $\lfloor \alpha Q \rfloor$ nodes with larger values as the storage node combination.

- Step 5: Population generation: generate the offspring population $S_t$ from the parent population $P_t$.

- Step 6: Form a combined population: $R_t = P_t \cup S_t$, according to the fast nondominated sorting approach, and select $N_P$ individuals as the next-generation parent population $P_{t+1}$ from the combined $2N_P$ individuals.

- Step 7: Go to Step 2 unless the termination criterion is met.

These steps can be represented as the flowchart, shown in Fig. 5.

## 5. Numerical Experiments

After implementing our proposed approach, we have carried out a series of numerical experiments using several networks and network instances. In this section, we summarize the main results and the performance of the proposed MOEEA. Since there is no multi-objective method yet to solve the problem of heterogeneous cost network disintegration, we can only choose some single-objective state-of-the-art disintegrate strategies and then calculate the cost. A series of comparative analyses have been carried out between MOEEA and five other classical and representative disintegrate strategies: the high degree first (HDF), the high degree adaptive (HDA), the high betweenness first (HBF), the high betweenness adaptive (HBA) and collective influence (CI) algorithms. These five methods are widely recognized for their more general applicability and good performance. Pareto fronts and performance measures have been calculated, and the results will be presented and discussed in this section.

Firstly, some real-world benchmark networks are used, including the Zachary Karate Club, the Contiguous States of USA, the Dolphins and the Jazz Musicians network. Secondly, three different model networks have

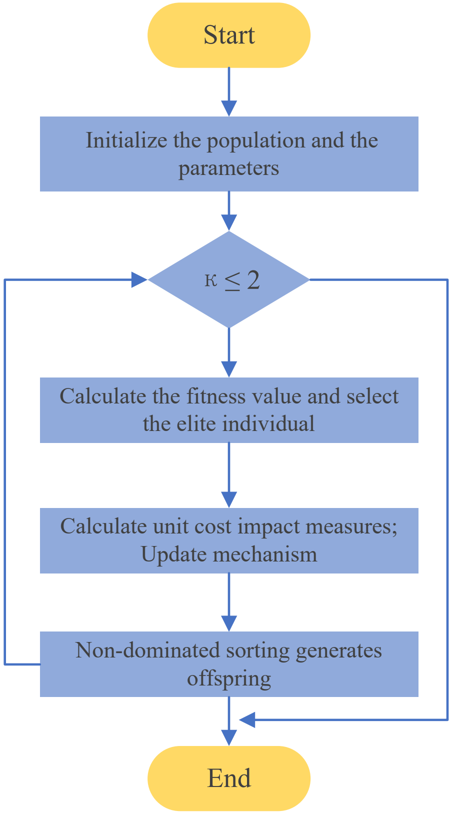

Start
Initialize the population and the parameters
κ ≤ 2
Calculate the fitness value and select the elite individual
Calculate unit cost impact measures; Update mechanism
Non-dominated sorting generates offspring
End

Figure 5: The flowchart of the proposed MOEEA.

also been applied to validate the effectiveness of the proposed MOEEA. In the simulations, the cost function has been assumed to be the exponential function, as discussed earlier in Section 3. Then, a further comparison and some evaluations have also been carried out so as to verify that the algorithm is still valid for the power cost functions.

All the experiments have been carried out on a computer, running Windows 10 with Intel i7-8700 CPU and 8GB RAMs, implemented using MATLAB 2019b. Details of numerical experiments and the results will be given soon after the discussion of the four benchmark networks.

### *5.1. Four benchmark networks*

To systematically evaluate our proposed algorithm, four different types of undirected, unweighted real-world benchmark networks have been used to compare the performance of the present algorithm with those of five

Table 1: Basic features of the four benchmark networks.

| Network | $N$ | $M$ | $k_{\max}$ |
|---|---|---|---|
| Karate Club | 34 | 78 | 17 |
| Contiguous States | 49 | 109 | 8 |
| Dolphin | 62 | 159 | 12 |
| Jazz Musicians | 198 | 2742 | 100 |

other strategies. These networks span different areas, including social networks, physical networks, communication networks, and collaborative networks. The used datasets are available at the KONET link.[1]

The well-known Zachary Karate Club network is a widely used benchmark. The original data was collected by Wayne Zachary [49], concerning a university karate club. This can be considered as a small-scale epitome of a real-world social network [44]. This data set is often used to find socially active people who maintain overall social relationships.

The Contiguous States of USA network [50] include 49 states of the United States of America, excluding the states of Alaska and Hawaii. The 48 contiguous states and the District of Columbia are not connected by land with the other states. In this network, an edge is used to denote that two states share a border. This benchmark data set is used to simulate and plan of some infrastructure construction, including the selection of large transmission stations, large logistics warehouses, transportation transit stations, and other locations.

The Dolphin network complied by David et al. [51, 52], which concerns the relationship between bottlenose dolphins living in the fjords of New Zealand. Edges represent their frequent associations and relationships, including making friends between males, raising their offsprings by females, and mating relationships.

Jazz Musicians network [53], as another benchmark, is considered as a collaboration network. Each jazz musician is represented as a node, and an edge represents a collaboration between two musicians. There are 198 musicians and 2742 edges in total.

Some basic statistical features of these four benchmark networks are shown in Table 1, including the number of nodes $N$, the number of edges $M$, and the maximum degree $k_{\max}$.

[1] http://konect.uni-koblenz.de/networks/

### *5.2. Experimental settings and result*

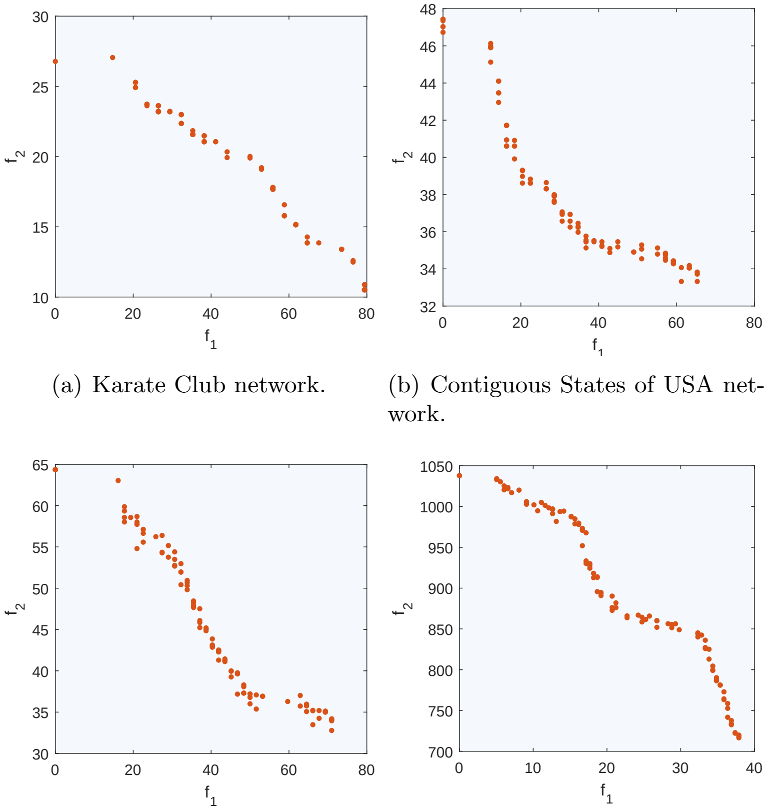
All the algorithms and different models are all implemented and run on the same computer. Each algorithm has been independently executed 20 times to reduce any possible dependence or influence of any initial configuration on the results. The population size of our MOEEA is set to 100, the total number of iterations is taken as 100. The percentage of reservation $\alpha = 0.2$ is used. Since the CI algorithm is not very sensitive to the adjustable parameter $\ell$ [12, 54], the ball radius $\ell$ is set to 3 in these experiments. For a given attack intensity, the critical condition of network disintegration $\kappa \leq 2$ is taken as the termination condition of theses five strategies: HDF, HDA, HBF, HBA and CI.

To better investigate the impact of real costs and evaluate the performance of the proposed algorithm, we have used the two cost functions defined by [19]. Firstly, the performance of different disintegration methods is studied when the cost is an exponential function as Eq. (4). The parameter $\beta$ is set to 0.4. The experiments are carried out in real networks and model networks, respectively. Then the results are analyzed in detail. Besides, the power-law cost function (Eq. (5)) is also used in our experiments, which intends to verify the universality of the proposed algorithm. The parameter $\gamma = 0.6$ is used. A comparison is carried out to see any possible influence of different cost functions on the final results.

The results of the experiments are visualized and analyzed by using the above parameter settings. As a multi-objective problem, under any attack intensity, a Pareto Front, instead of a single solution, has been obtained. This is the significant difference between the existing single-objective optimization methods.

As examples, we have used the attack intensity $Q = 7$, $Q = 17$, $Q = 18$, $Q = 125$ for the Karate Club network, Contiguous States of USA, Dolphin network and Jazz Musician network, respectively. Their Pareto Fronts are shown in Fig. 6.

The advantage of our bi-objective approach is that it not only solves the problem but also gives decision-makers more choices. There may be some nodes that are not easy to control in practice, thus alternative options and choices are desirable. For example, a transmission power station may be located in an area where transportation is not convenient, and the key suspects in a criminal network cannot be identified. In such scenarios, different alternatives should be considered and evaluated. Also, the elite individual mentioned above might be the final solution when other choices are not obvious.



(a) Karate Club network.

(b) Contiguous States of USA network.

(c) Dolphins network.

(d) Jazz Musicians network.

Figure 6: The Pareto Fronts of four benchmark networks.

### *5.3. The experimental results of Exponential function*

For a fair comparison of different methods, we use the final elite individual as the 'optimal solution' of MOEEA, because the other five disintegration strategies can only get one solution at a time. As the other methods are all single-objective problems, we compare the two objectives separately: 1) compare the extent of damage to the network $S(Q)$; 2) the cost of the solution found by the algorithm.

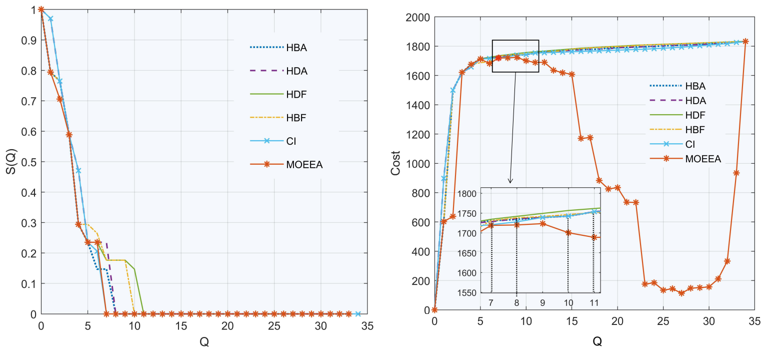


Figure 7: Comparison of experimental results of the Zachary Karate club network with different methods.

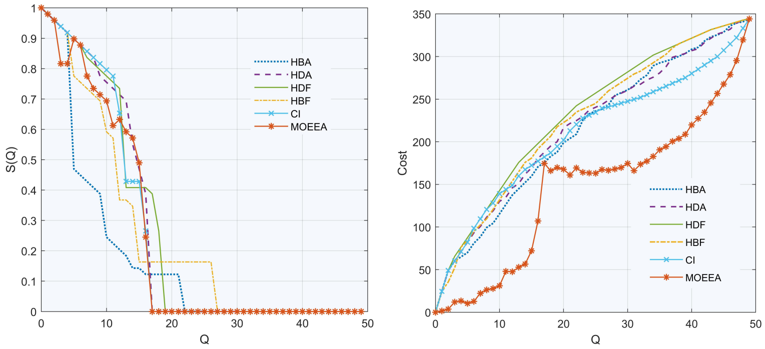


Figure 8: Comparison of experimental results of the Contiguous States network with different methods.

These features mean that two graphs are presented for each network. The experimental results are summarized in Fig. 7 to Fig. 10. For the Karate Club network, Fig. 7 shows the results of different methods. From these two figures, we can see that, compared with HBA, HDA, HDF, HBF and CI, the algorithm proposed in this paper performs more competitively.

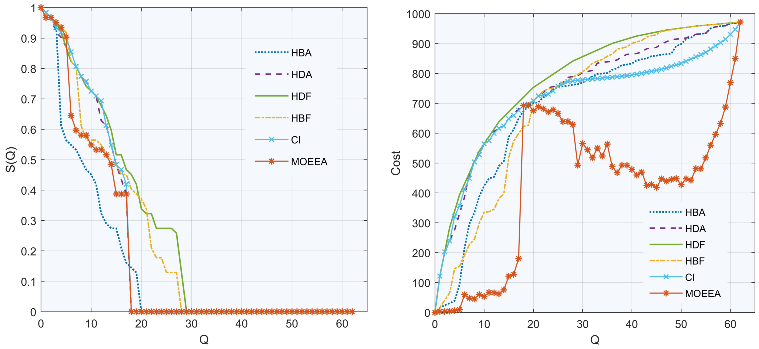


Figure 9: Comparison of experimental results of the Dolphins network with different methods.

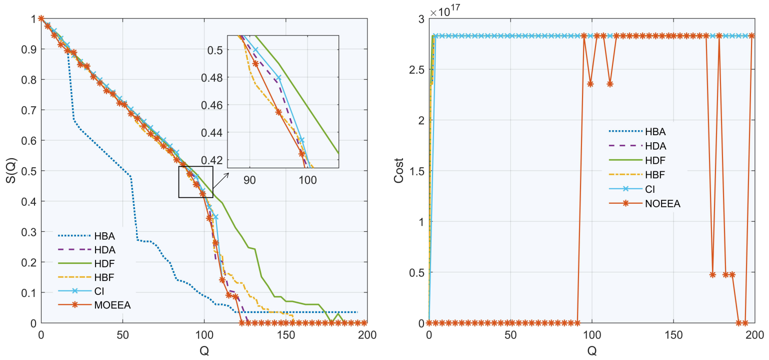


Figure 10: Comparison of experimental results of the Jazz Musicians network with different methods.

The left figure has shown that MOEEA can destroy the network with fewer key nodes than other methods. The cost curves of these methods in the figure on the right show that the proposed method can find solutions with lower costs. After the network has disintegrated, as the increase of attack intensity, some low-cost attack strategy may be found by MOEEA. As a matter of fact, when the attack intensity is small, in order to disintegrate the network, the key nodes found by the algorithm often have high costs. With the increase of attack intensity, some node combinations gradually play a more critical role in the network disintegration.

For the Contiguous States network, the results and the comparison of different strategies are summarized in Fig. 8. The critical attack strength ($S(Q)$ first approaches to zero) $Q_c$ for MOEEA is 17, while $Q_c$ is equal to

19, 22, 27 for HDF, HBA, HBF, respectively. MOEEA can disintegrate the network with a minimum number of nodes (attack performance is the same as HDA and CI). From the lower cost curves, we can still see that, under any attack intensity, the cost of MOEEA is the lowest among six strategies. Therefore, MOEEA is more effective than the other five attack strategies.

For the Dolphins network, the results are shown in Fig. 9. When the network is in a critical disintegration state, the value of $Q$ reaches the critical strength $Q_c$. As presented on the left of the figure, the $Q_c$ for our proposed MOEEA is also lower than or the same as that obtained by other five strategies. When the attack intensity is $Q_c$, our method is slightly higher in cost than HBA and HBF, but the network has not yet disintegrated at this time for these two methods. Furthermore, except for the attack intensity $Q_c$ (for MOEEA), the cost of the attack nodes found by our method is far less than that of the other five strategies.

As presented in Fig. 10, for the Jazz Musicians network, HBA cannot dismantle the network. In terms of network disintegration, our method still has some advantages, and in terms of cost, MOEEA can only find better attack options at certain attack intensity. One reason is the close connection within the Jazz Musicians network. The maximum degree of a node is $k = 100$, and the cost of this single node is $\exp(\beta k) = \exp(40) \approx 2.3538 \times 10^{17}$. Therefore, other nodes with low cost will have much less impact on the results.

Table 2: Cost comparisons of different methods for network collapse.

| Networks | | HBA | HDA | HDF | HBF | CI | MOEEA |
|---|---|---|---|---|---|---|---|
| Karate Club | $Q$<br>Cost | 8<br>1.734e+03 | 8<br>1.736e+03 | 11<br>1.762e+03 | 10<br>1.747e+03 | 7<br>1.721e+03 | 27<br>113.185 |
| Contiguous States | $Q$<br>cost | 22<br>2.089e+02 | 17<br>1.877e+02 | 19<br>2.200e+02 | 27<br>2.449e+02 | 17<br>1.822e+02 | 21<br>160.984 |
| Dolphins | $Q$<br>cost | 20<br>7.011e+02 | 18<br>6.924e+02 | 29<br>7.956e+02 | 28<br>7.882e+02 | 18<br>6.924e+02 | 41<br>408.813 |
| Jazz Musicians | $Q$<br>cost | -<br>- | 125<br>2.829e+17 | 182<br>2.829e+17 | 158<br>2.829e+17 | 123<br>2.829e+17 | 190<br>7.347e+12 |

Table 3: The comparison of min-cost of MOEEA with other methods.

| networks | HBA | HDA | HDF | HBF | CI |
|---|---|---|---|---|---|
| Karate | ↓ 93.47% | ↓ 93.48% | ↓ 93.58% | ↓ 93.52% | ↓ 93.42% |
| States | ↓ 22.94% | ↓ 14.23% | ↓ 26.83% | ↓ 34.27% | ↓ 11.64% |
| Dolphins | ↓ 0.4169% | ↓ 40.96% | ↓ 48.62% | ↓ 48.13% | ↓ 40.96% |
| Jazz | ↓ 100% | ↓ 100% | ↓ 100% | ↓ 100% | ↓ 100% |

For further analyses, some experimental details are summarized in Table 2 and Table 3. The minimum costs of HBA, HDA, HBF, HDF, CI and MOEEA for network disintegration are shown in Table 2. It can be seen that the proposed MOEEA can dismantle the network at the lowest cost.

When the Karate network attacks 27 nodes, the cost is 113.1850, which is much lower than the CI algorithm's 1720.7. The same thing happened on the other three networks.That means the performance of MOEEA is better than other disintegration strategies on these real networks.

More specifically, the other five methods are considered as baseline methods, the percentage of cost reduction of our method is shown in Table 3. Compared with other methods, it can be seen that the cost of our method is reduced by at least 11.64%. For the Jazz network, the cost found by MOEEA is far less than other five algorithms. In summary, the MOEEA can disintegrate the networks with the minimum cost.

### *5.4. The experiment of three model networks*

The above networks are real small-scale networks. In this subsection, we test three widely used model networks such as stochastic networks, small-world networks and power-law networks. The Erdös and Rényi (ER) network is the first stochastic network model [55], and the distribution of the connections among nodes obeys the Poisson distribution. The Watts Strogatz (WS) small-world network model has high clustering [56]. The Barabasi Albert (BA) is a scale-free network model [26], which exhibits a power-law distribution of degrees of nodes. Here, we have used these three model networks to test the performance of our proposed MOEEA.

For the WS network, its average degree $m$ and the removal probability $p$ are 4 and 0.5, respectively. For the BA network, the number of initial nodes ($m_0$) is set to 3, the removal probability $p = 0.8$ is used, and the average degree ($m_0$) is taken as 5. We used $p = 0.02$ for the ER network. The number of nodes $N$ of these model networks is 1000. The population number $N_P$ of MOEEA is set to 100, and supposed the total number of iterations is 100. The cost function takes the exponential function mentioned above as an example, and the results are visualized in Fig. 11 to Fig. 13.

Again from these two figures, we can see in general that the MOEEA shows better performance than the other five attack strategies. For the BA network, when the attack strength is greater than 380, the lowest cost of MOEEA is $1.4009E{+}13$, which is far less than the cost $2.2878E{+}17$ of other methods. For the ER network, the lowest cost of MOEEA is $1.3964E + 07$, which is less than the cost of other methods. In other words, the proposed MOEEA requires a lower cost than other methods except for part of attack strength (close to the cost of other methods) on a variety of model networks.

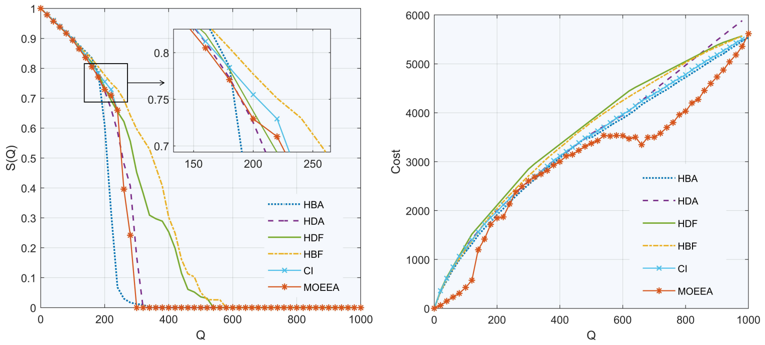


Figure 11: Comparison of experimental results of the WS small-world network with different methods.

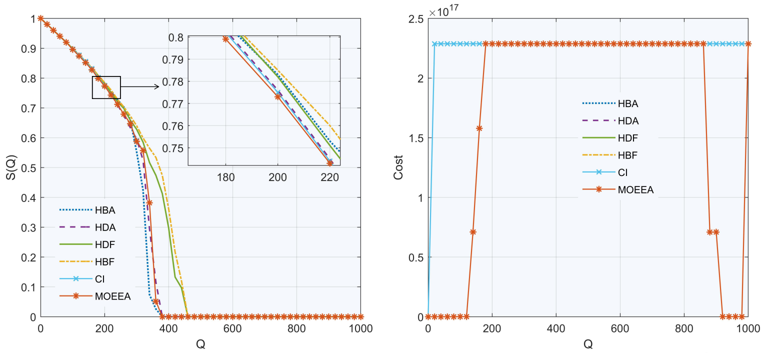


Figure 12: Comparison of experimental results of the BA network with different methods.

### 5.5. The experiment of Power-law cost function

The experiments in this section investigate the universality of the proposed algorithm and explore the possible influence of different cost functions on the six strategies. To compare the results for four real networks, we have carried out some further experiments by using another cost function: the power-law cost function, which is defined as $c(k) = k^{\gamma}$ and $\gamma = 0.6$. All other parameters of algorithms remain the same as before in the previous experiments. The comparison results of different algorithms are shown in Fig. 14 to Fig. 17.

Similar to the exponential cost function, MOEEA still shows a better performance than other methods when the cost is a power function. CI strategy can usually break up the network with fewer nodes for other five comparison algorithms, and its attack performance is generally better than

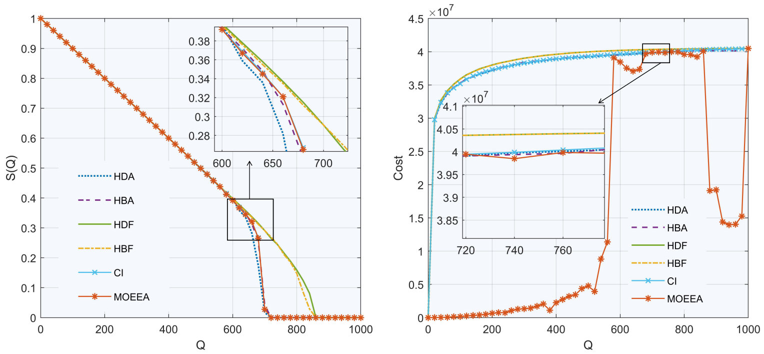


Figure 13: Comparison of experimental results of the ER network with different methods.

those of HBA, HDA, HDF and HBF. Its attack cost is slightly better than the other four algorithms. Therefore, we focus on the comparison between our algorithm and CI.

From Fig. 14, we can see that, when the attack strength is less than $Q_c$, MOEEA can find a more effective key node-set than CI strategy, which can be proved from the lower $S(Q)$ curve. In addition, we can see that the cost curve of MOEEA is below the curve of CI. In other words, the cost of the solution obtained by MOEEA is smaller than that of CI. The same is true for the Contiguous States network, Dolphin network, and Jazz Musicians network. (Only under some specific attack strength, the cost of MOEEA is close to that of other methods. This is because there are some nodes with high degree values in the network, and their costs are much higher than other nodes. If these nodes are included in the key node set, the cost will be relatively high.) Moreover, since MOEEA is a bi-objective algorithm, under the same attack strength, we will get a set of Pareto optimal solutions. If it is not required to completely disintegrate the network (there can still be smaller connected branches), the costs can be even lower.

Furthermore, the $Q_c$ value and minimum cost of these six strategies are given in Table 4. The cost values of these six methods become lowest when the value of $Q$ corresponds to the critical attack intensity $Q_c$. The costs of the MOEEA are the lowest for the Karate Club network and the Contiguous States network. For the Dolphins network and Jazz musicians network, the cost values for MOEEA are also lower than those for HBA, HDA, HDF, HBF, and the same as that obtained by CI.

It is worth pointing out there is some difference from preceding results using the exponential function. The reason may be that two different nodes,

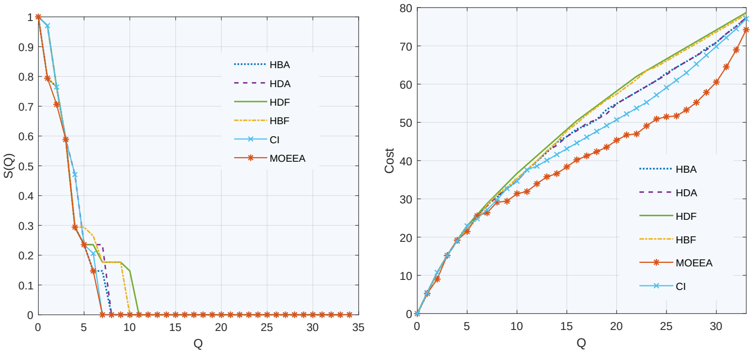


Figure 14: Comparison of experimental results of the Zachary Karate Club network with different methods.

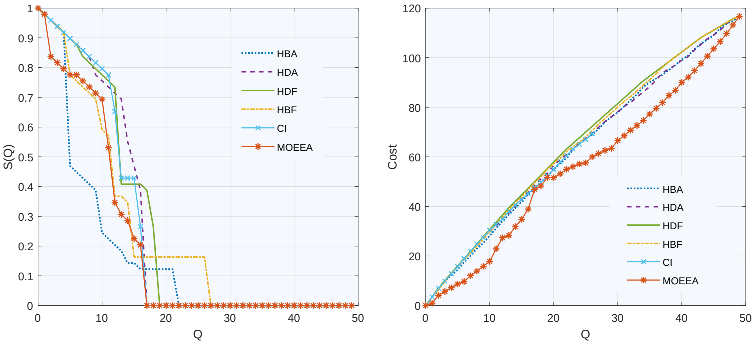


Figure 15: Comparison of experimental results of the Contiguous States network with different methods.

through a cost function (power function), map the originally two very different degree values to relatively close cost values. For example, for the Jazz Musician network, the removal cost of the node with $k = 100$, using the exponential function, is $2.3538 \times 10^{17}$ as we mentioned before. However, the cost in terms of the power-law function $k^{\gamma}$ is $100^{0.6} = 15.8489$. Thus, the costs by the power-law function are much lower and are relatively close to each other. So, the influence of the cost objective on the final result of the algorithm becomes smaller.

The comparison of the lowest cost value of the MOEEA with the values of other methods are summarized in Table 5 where the values are the cost reductions in percentage. The minimum cost of disintegrating the network found by MOEEA is almost always lower than other methods, although the percentages of reductions vary. The results of CI are mostly consistent

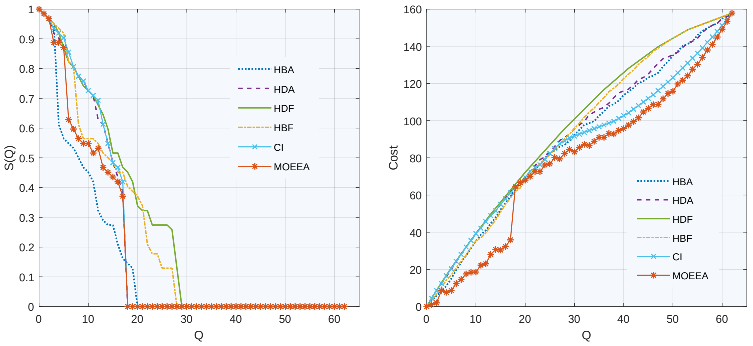


Figure 16: Comparison of experimental results of the Dolphins network with different methods.

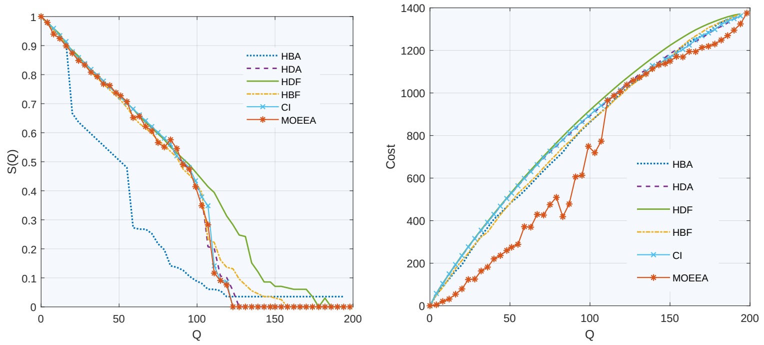


Figure 17: Comparison of experimental results of the Jazz Musicians network with different methods.

with that of proposed MOEEA, which also shows that the solution found by our algorithm is feasible. The above experiments and comparison show that our proposed MOEEA can find a set of min-cost optimal key nodes to disintegrate the networks.

## 6. Algorithm analysis

### *6.1. Parametric studies*

In MOEEA, an elitist strategy is used and the combined information of nodes is implemented based on parameter $\alpha$. It can be considered as the percentage of influence of the elite individual on the new offsprings, and $\alpha$ takes values in the range $(0, 1)$. In order to explore the effect of the value of $\alpha$ on the performance of the algorithm, we have used three different types

Table 4: Cost comparison of different methods for network collapse.

| Fun | | HBA | HDA | HDF | HBF | CI | MOEEA |
|---|---|---|---|---|---|---|---|
| Karate Club | N<br>Cost | 8<br>30.401 | 8<br>30.765 | 11<br>38.948 | 10<br>35.325 | 7<br>27.142 | 7<br>26.360 |
| Contiguous States | N<br>cost | 22<br>59.620 | 17<br>48.437 | 19<br>55.239 | 27<br>73.184 | 17<br>47.430 | 17<br>46.822 |
| Dolphins | N<br>cost | 20<br>68.979 | 18<br>64.359 | 29<br>98.278 | 28<br>90.157 | 18<br>64.359 | 18<br>64.359 |
| Jazz Musicians | N<br>cost | -<br>- | 125<br>1063 | 182<br>1347.8 | 158<br>1222.8 | 123<br>1037.9 | 123<br>1037.9 |

Table 5: The comparison of the lowest cost of MOEEA with other methods.

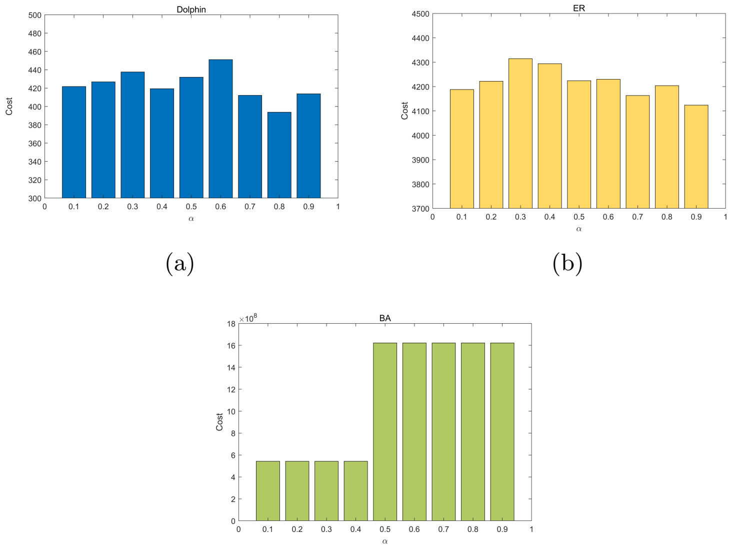
| networks | HBA | HDA | HDF | HBF | CI |
|---|---|---|---|---|---|
| Karate | ↓ 13.29% | ↓ 14.32% | ↓ 32.32% | ↓ 25.38% | ↓ 2.80% |
| States | ↓ 21.47% | ↓ 3.34% | ↓ 15.24% | ↓ 36.02% | ↓ 1.29% |
| Dolphins | ↓ 6.70% | −0% | ↓ 34.51% | ↓ 28.61% | −0% |
| Jazz | - | ↓ 2.36% | ↓ 22.99% | ↓ 15.12% | −0% |

of networks (all of which have 300 nodes) and compared the performance of the algorithm for three networks, respectively.

In our experiments, the value of $\alpha$ varies from 0.1 to 0.9 with the increment of 0.1. First, we have used the Dolphin network for parametric study. Then, we have used two model networks (i.e., ER and BA) for testing. Five different network instances are selected for each type of network. For the convenience of algorithm comparison, we have chosen a certain attack strength for each network at which the network can be completely disintegrated (i.e., $S(Q)$ equals 0), which allows us to compare the minimum attack costs found by the algorithms. The results of the experiments are shown in Fig. 18.

For Dolphin networks, when the attack strength is 49 ($Q = 49$), the results of different values of $\alpha$ are shown in Fig. 18(a). It indicates that when $\alpha = 0.8$ for Dolphin network, the 'Cost' is the lowest. When $\alpha = 0.1, 0.4, 07, 0.9$, the performance of MOEEA is also very good. As seen in Fig. 18(b), for ER network, the attack cost is the best when $\alpha = 0.9$. Fig. 18(c) implies that the performance of MOEEA is sensitive to the value of $\alpha$ for BA networks. When $\alpha$ is less than or equal to 0.4, the disintegration cost is relatively low. In general, the performance of the algorithm decreases when the value of $\alpha$ increases. Due to such influence of $\alpha$, the value of $\alpha$ in this paper is chosen randomly, and it can be expected that choosing the right $\alpha$ value may give better results.

In addition, the parameters $\beta$ and $\gamma$ in the cost functions (i.e., exponential function and power law) are chosen arbitrarily in the paper, and the focus is on the performance of the algorithm for solving network disintegration problems with heterogeneous costs. To investigate the effect of different cost functions on the algorithm, we discuss the values of the parameters of



(a)

(b)

(c)

Figure 18: The experimental results of Dolphin, ER and BA networks with different population sizes: (a) Dolphin network, (b) ER network, (c)BA network.

these cost functions.

Theoretically, the parameters $\beta$ and $\gamma$ can take any real number lager than 0. However, when their values are less than 0.1, the effect of cost is very small, which is similar to the network disintegration problem with equivalent cost. So we set the values of the parameters to vary as 0.1, 0.5, 1 and 1.5. Taking the small-world networks and ER network as examples, the network sizes are set to 300 and five network instances have been selected randomly for each type model. Other parameters in the algorithm remain unchanged. In order to facilitate a fair comparison, the minimal cost of the key nodes which make the network disintegration is calculated. The experiment results are summarized in Table 6 to Table 9.

Table 6: The experimental results of ER networks with different values of $\beta$.

| $\beta$ | HBA | HDA | HBF | HDF | CI | MOEEA |
|---|---|---|---|---|---|---|
| 0.1 | 167.00 | 159.07 | 244.53 | 263.03 | 158.38 | 155.40 |
| 0.5 | 1.35e+03 | 1.33e+03 | 1.88e+03 | 1.99e+03 | 1.30e+03 | 1.28e+03 |
| 1 | 2.35e+04 | 2.34e+03 | 2.99e+03 | 3.08e+04 | 2.23e+04 | 2.22e+04 |
| 1.5 | 5.13e+05 | 5.09e+05 | 5.95e+05 | 6.03e+05 | 4.74e+05 | 4.39e+05 |

For the exponential function, the results of SW network and ER network with different values of parameter $\beta$ are shown in Table 6 and Table 7. It

Table 7: The experimental results of ER networks with different values of $\beta$.

| $\beta$ | HBA | HDA | HBF | HDF | CI | MOEEA |
|---|---|---|---|---|---|---|
| 0.1 | 309.15 | 404.56 | 301.17 | 405.43 | 301.46 | 301.46 |
| 0.5 | 1.26e+04 | 1.57e+04 | 1.34e+04 | 1.57e+04 | 1.34e+04 | 9.41e+03 |
| 1 | 3.80e+06 | 4.44e+06 | 4.31e+06 | 4.44e+06 | 4.31e+06 | 9.61e+05 |
| 1.5 | 2.36e+09 | 2.68ee+09 | 2.67e+09 | 2.68e+09 | 2.67e+09 | 1.20e+08 |

can be seen that MOEEA can find the node set with the minimum cost under different parameter ranges. As with the increase of parameter $\beta$, the advantage of our algorithm becomes more significant, which shows once again that our algorithm is better when the cost heterogeneity is stronger.

Table 8: The experimental results of SW networks with different values of $\gamma$.

| $\gamma$ | HBA | HDA | HBF | HDF | CI | MOEEA |
|---|---|---|---|---|---|---|
| 0.1 | 119.23 | 112.55 | 175.39 | 189.29 | 112.47 | 110.17 |
| 0.5 | 223.76 | 213.57 | 328.87 | 353.88 | 212.73 | 208.73 |
| 1 | 497 | 480 | 727 | 779 | 476 | 468 |
| 1.5 | 1.12e+03 | 1.09e+03 | 1.62e+03 | 1.73e+03 | 1.08e+03 | 1.07e+03 |

Table 9: The experimental results of ER networks with different values of $\gamma$.

| $\gamma$ | HBA | HDA | HBF | HDF | CI | MOEEA |
|---|---|---|---|---|---|---|
| 0.1 | 146.53 | 139.45 | 183.55 | 190.88 | 139.50 | 134.71 |
| 0.5 | 328.28 | 314.33 | 413.46 | 429.63 | 314.89 | 304.96 |
| 1 | 915 | 881 | 1153 | 1196 | 884 | 859 |
| 1.5 | 2.60e+03 | 2.51e+03 | 3.25e+03 | 3.37e+03 | 2.52e+03 | 2.46e+03 |

For the power law function, Table 6 and Table 7 show the results of the SW network and ER network with different values of parameter $\gamma$. The performance of the proposed algorithm is slightly better than the other five algorithms no matter what the parameter value is. Therefore, we can conclude that MOEEA is suitable for solving network disintegration problem with different cost functions. In addition, the performance of the proposed algorithm is better than these state-of-the-art algorithms.

### *6.2. Algorithmic convergence and complexity*

For multi-objective problems, to assess the results obtained by the proposed algorithm, the measurement error distance has been used as a performance indicator [57]. This error distance is defined as

$$\|PF - PF^{*}\| = \sum_{i=1}^{N} \min(d(\vec{x}_i, v)),\ v \in PF^{*},\ x_i \in PF.$$

where $d(\vec{x}_i, v)$ means the Euclidean distance between $x_i$ and the point $v$ in the baseline Pareto front $PF^{*}$. Here, $PF$ is the current Pareto Front found by the algorithm. The higher the value of $\|PF - PF^*\|$, the greater the deviation of the found solution set from the baseline $PF^*$.

Taking the Karate Club network as an example, the maximum number of iterations is set to 2000, and the Pareto Front obtained is considered as the baseline $PF^*$. Then the Pareto Front is recorded every 10 iterations. By plotting out the error distances during iterations, the variation of convergence rates or the convergence property can be seen. As shown in Fig. 19, the convergence curve indicate that the algorithm converges relatively fast at first, which means that the MOEEA can find relatively good solutions by few iterations. After the number of iterations reaches 200, the convergence rate becomes slower and the algorithm gradually converges to a stable set of solutions.

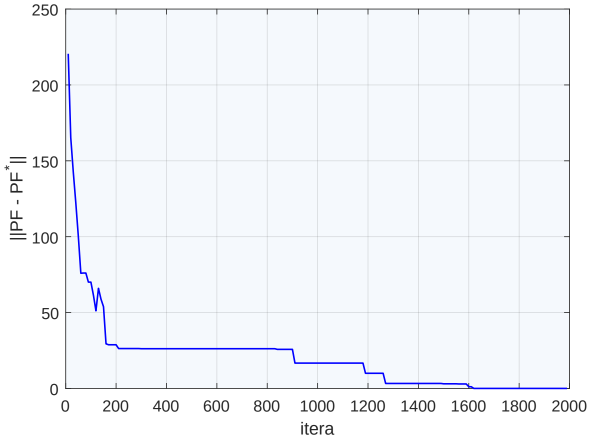


Figure 19: Convergence curve of MOEEA.

Now we analyze the complexity of the proposed algorithm. During each iteration, the unit cost importance measure of the target attack node is calculated first. Since we only needs to calculate $UIM$ values of the target attack nodes in elite individual, the complexity is $O(Q * \langle k \rangle)$, where $Q$ is the attack intensity, $\langle k \rangle$ is the average degree of nodes in the network. Then, these $Q$ values are sorted using a fast sort method with complexity $O(Q \log Q)$.

In addition, the complexity of the update strategy is low. The complexity of fitness evaluations depends mainly on the calculation of the maximum connected cluster. Using the Depth First Search (DES) to calculate the largest connected component, the complexity is $O(N^2)$. The algorithm complexity of non-dominated sorting is $O(2 * (N_p)^2)$ where $N_p$ is the population size of the algorithm. Therefore, the complexity of each iteration is $O(Q \langle k \rangle + Q \log Q + (N_p) * N^2 + 2 * (N_p)^2)$ , That is, the complexity of the algorithm is at most $O(N_p * N^2 + (N_p)^2)$. When the number of nodes is much larger than the population size, the overall complexity of our proposed method is $O(N^3 T)$, where $T$ is the maximum number of iterations.

### *6.3. Strength and weakness of MOEEA*

In this paper, the network disintegration problem with heterogeneous cost is solved as a multi-objective problem, and the cost factor is truly included in the entire optimization process. Experiments have shown that our proposed MOEEA can solve this problem well. Now we can discuss the advantages and disadvantages of our algorithm.

Our MOEEA is the first attempt of its kind to use a multi-objective metaheuristic algorithm to solve the minimum cost network disintegration problem. We propose a unit cost importance measure ($UIM$), which combines the cost and the importance of node and facilitates the selection and comparison of nodes in subsequent algorithms. The elitist strategy not only preserves the best combination information of nodes to guide the search process of subsequent algorithms, but also stores and updates the 'optimal solution' for each generation. And our algorithm only needs to calculate $UIM$ values of the target attacking nodes in the elite individual in each iteration, so the calculation is not large. In addition, in the update strategy, the number of 'genes' of a parent is randomly generated in each iteration; that is, the local search and global search of the algorithm are dynamically adjusted. It not only takes into account the convergence speed of the algorithm, but also avoids the algorithm from falling into the local optimum. Experimental results show that the MOEEA tends to find low-cost disintegration solutions and the multiple Pareto optimal solutions obtained can provide more choices for decision makers.

The computational cost of the multi-objective metaheuristic algorithm is relatively high, but it can be improved through some algorithm parallelization and modification, which will form an important topic for future research.

## 7. Discussions

### *7.1. Illustrative Example for Visualization*

In order to visualize the results of the algorithm, a rather small-scale network: Askcal network, which is the communication network of Student

Government in the University of Ljubljana [58] is used as an example to observe the differences of different methods. The network contains 11 nodes and 29 edges, and its connection mode is shown in the Fig. 20.

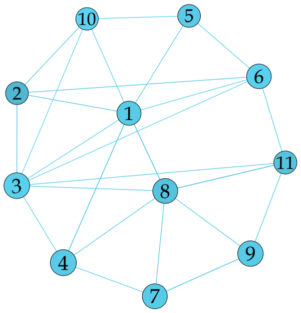
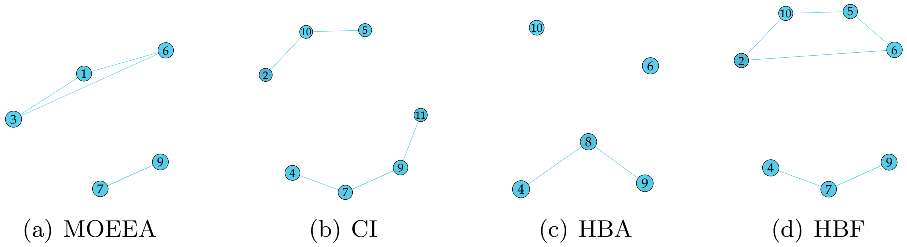


Figure 20: Askcal network.

The power law function is used as the cost function. The experimental results of the six algorithms are summarized in Table 10. It can be seen that among these six algorithms, MOEEA finds the key nodes with the lowest cost in the state of network disintegration. The corresponding network nodes status are shown in Fig. 21 where all networks have lost most of their connectivity.

Table 10: The comparisons of five algorithms with Askcal network.

| | MOEEA | CI | HBA | HDA | HBF | HDF |
|---|---|---|---|---|---|---|
| S(Q) | 0 | 0 | 0 | 0 | 0 | 0 |
| Cost | 34.16 | 51.30 | 49.44 | 51.30 | 48.87 | 51.30 |



(a) MOEEA (b) CI (c) HBA (d) HBF

Figure 21: The state of the network after being attacked.

In fact, the members in this communication network are so closely connected that the presence or absence of a few nodes does not affect the structure of the whole network. While the degree-based algorithms may fail, the proposed algorithm can still perform well.

### *7.2. The Role of Preprocessing*

For simulating large-scale networks, there are two major challenges: storage and computing efforts. Such challenges are also relevant in the big data processing. Data reduction is usually used, that is, reducing the size of the data without affecting (or at least not substantially affecting) the final result. Therefore, it makes sense to preprocess the network before the algorithm starts.

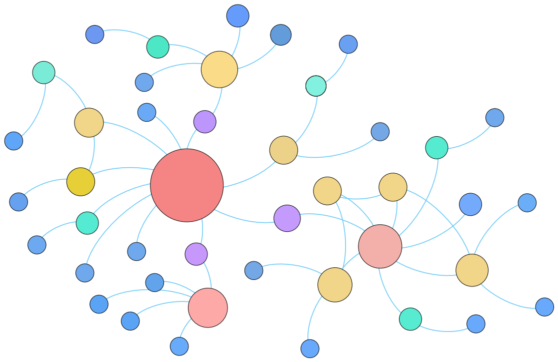


Figure 22: The schematic diagram for preprocessing.

Obviously, for a real-world network, not all nodes are equal. Some nodes have little or no influence on the overall structure and performance of the network, and these nodes are unlikely to be the key nodes we are looking for. In fact, almost all networks can have isolated nodes and peripheral nodes (their degree is 1, and such nodes are marked in blue in Fig. 22). Such nodes have little impact on the collapse of the network and should be removed first as part of preprocessing. For medium-sized networks, such preprocessing is usually sufficient. For large-scale networks, we should continue to reduce their sizes. More information is needed. From Fig. 22, it can be seen that there are six nodes of degree 2 (marked in aquamarine colours) with one edge connected to a peripheral node (marked with blue, or blue nodes), and the other edge is only connected to one other node in the network. Attacking such a 2-degree (aquamarine) node will only separate the two nodes from the entire network, and thus has little effect on the network structure.

Hence, the degree of these nodes that need to be removed next for preprocessing is 2. It is worth noting that not all nodes with a degree of 2 in the network should be deleted. Because some nodes with degree 2 are critical to the connection of two important branches in the network, such as the three nodes marked in purple shown in Fig. 22. These nodes may be weakly connected, but they are key nodes, which can easily be neglected in the existing centrality-based methods. Therefore, in the preprocessing step, the node with degree 1 (blue nodes) should be deleted first; then, after the first step, if the number of neighbors of the node with original degree 2 becomes 1, these nodes should be deleted. The pseudocode for such preprocessing can be summarized in Algorithm 2.

**Algorithm 2** Preprocessing the network.

```
Input: the adjacency matrix $A$
$A_{temp} \leftarrow A$
$\mathbf{d} \leftarrow$ degree of nodes // calculate by $A_{temp}$, $\mathbf{d}$ is a $1 \times N$ vector.
$\mathbf{d}_{temp} \leftarrow \mathbf{d} - \mathbf{1}_{1\times N}$
$remove \leftarrow$ the set of -1 and 0 bits in $\mathbf{d}_{temp}$
$A_{temp}(remove, :) \leftarrow \mathbf{0}$ // delete corresponding rows
$A_{temp}(:, remove) \leftarrow \mathbf{0}$ // delete corresponding columns
$\mathbf{d}_{new} \leftarrow$ degree of nodes // recalculate by $A_{temp}$
$remove \leftarrow$ the set of 1 bits in $\mathbf{d}_{new}$
$A_{temp}(remove, :) \leftarrow \mathbf{0}$ //delete corresponding rows
$A_{temp}(:, remove) \leftarrow \mathbf{0}$ //delete corresponding columns
return $A_{temp}$ as the new adjacency matrix
```

The purpose of preprocessing is to reduce the size of the network and to make the calculations faster. Taking the ER network with 1000 nodes as an example, the parameter $p$ is 0.003, $\gamma$ is set to 0.4. For the convenience of comparison, assuming that the attack strength is 300, both the population size and the maximum number of iterations are set to 100. The running time and cost of MOEEA with preprocessing are shown in the Table 11.

Table 11: Comparison of pretreatment effect.

| | time(s) | cost |
|---|---|---|
| preprocessing + MOEEA | 43.89 | 519.67 |
| MOEEA | 75.08 | 519.81 |

Our simulation shows that preprocessing can greatly accelerate the convergence speed of the algorithm. The good thing is that preprocessing has little influence on the selection of the key nodes in the network; that is, it does not affect the final solution found by the algorithm. This step can usually reduce the size of a large-scale problem significantly. It is worth noting that if the network scale is not very large, there is no need to do such preprocessing and we can skip this step.

## 8. Conclusions and Further Research

Research on network disintegration can help manage and control large-scale networks, such as transport networks, smart grids, infectious disease control, and others. Heterogeneous cost network attack problems may be relevant to many practical problems. In fact, most attack algorithms are very inefficient when considering degree-based attack costs.

In this paper, network disintegration with the heterogeneous cost is modelled as bi-objective optimization for the first time. We have proposed a global optimizer (metaheuristic algorithm), called elitism-based multi-objective evolutionary algorithm (MOEEA), which takes into account the heterogeneous cost of nodes. By defining a new unit cost importance measure (UIM), the importance of nodes and their removal cost are combined as an indicator, which makes it possible to select low-cost and efficient nodes. The interaction between nodes is considered, and part of the combination information of nodes is transmitted to the offspring of the population through elite individuals to guide the search process of the algorithm. In addition, both the preprocessing and ingenious update mechanism make the algorithm show better performance. We have compared our proposed MOEEA method with five state-of-the-art attack strategies (HDF, HBF, HDA, HBA and CI). The experimental results show that MOEEA can find lower cost disintegration solutions for the four benchmark networks and three stochastic model networks. When the heterogeneity of network nodes is higher, the effect of our method is more significant.

The results presented in this paper are some preliminary results. Still, they do indicate that a multi-objective optimization approach can provide better strategies than those by existing single-objective optimization approaches. Therefore, it would be useful to extend the current approach to investigate larger-scale real-world networks. In addition, further improvements and modifications can be explored to see if we can further improve the performance of the algorithm. Furthermore, it would be fruitful to extend the present algorithm to solve higher-dimensional networks with real costs incorporated in the mathematical formulations.

## Declarations of Interest

None.

## References

[1] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, S. Havlin, Catastrophic cascade of failures in interdependent networks, Nature 464 (2010) 1025–1028. https://doi.org/10.1038/nature08932.

[2] H. Chan, L. Akoglu, H. Tong, Make it or break it: Manipulating robustness in large networks, in: Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM, pp. 325–333.

[3] T. Tanizawa, G. Paul, R. Cohen, S. Havlin, H. E. Stanley, Optimization of network robustness to waves of targeted and random attacks., Physical Review E Statistical Nonlinear & Soft Matter Physics 71 (2005) 047101. https://doi.org/10.1103/physreve.71.047101.

[4] B. Shargel, H. Sayama, I. R. Epstein, Y. Bar-Yam, Optimization of robustness and connectivity in complex networks, Physical Review Letters 90 (2003) 068701. https://doi.org/10.1103/physrevlett.90.068701.

[5] L. A. Adamic, Power-law distribution of the world wide web, Science 287 (2000) 2115. http://doi.org/10.1126/science.287.5461.2115a.

[6] R. Cohen, K. Erez, D. Benavraham, S. Havlin, Resilience of the internet to random breakdowns, Physical Review Letters 85 (2000) 4626–4628. https://doi.org/10.1515/9781400841356.507.

[7] A. Braunstein, L. DallAsta, G. Semerjian, L. Zdeborová, Network dismantling, Proceedings of the National Academy of Sciences 113 (2016) 12368–12373. https://doi.org/10.1073/pnas.1605083113.

[8] L. Zdeborová, P. Zhang, H.-J. Zhou, Fast and simple decycling and dismantling of networks, Scientific reports 6 (2016) 37954. https://doi.org/10.1038/srep37954.

[9] D. J. Watts, A simple model of global cascades on random networks, Proceedings of the National Academy of Sciences of the United States of America 99 (2002) 5766–5771. https://doi.org/10.1515/9781400841356.497.

[10] L. Jain, R. Katarya, Discover opinion leader in online social network using firefly algorithm, Expert Systems with Applications 122 (2019) 1–15. https://doi.org/10.1016/j.eswa.2018.12.043.

[11] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, H. A. Makse, Identification of influential spreaders in complex networks, Nature Physics 6 (2010) 888–893. https://doi.org/10.1038/nphys1746.

[12] F. Morone, H. A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (2015) 65–68. https://doi.org/10.1038/nature14604.

[13] D. C. Bruno Requiao, G. A. Juan Carlos, G. Sebastian, Fast fragmentation of networks using module-based attacks, PloS one 10 (2015) . https://doi.org/10.1371/journal.pone.0142824.

[14] F. Morone, B. Min, L. Bo, R. Mari, H. A. Makse, Collective influence algorithm to find influencers via optimal percolation in massively large social media, Scientific reports 6 (2016) 30062. https://doi.org/10.1038/srep30062.

[15] X.-L. Ren, N. Gleinig, D. Tolić, N. Antulov-Fantulin, Underestimated cost of targeted attacks on complex networks, Complexity 2018 (2018) 1–15. https://doi.org/10.1155/2018/9826243.

[16] Y. Yang, T. Nishikawa, A. E. Motter, Small vulnerable sets determine large network cascades in power grids, Science 358 (2017) eaan3184. https://doi.org/10.1126/science.aan3184.

[17] Y. Deng, J. Wu, Y. J. Tan, Optimal attack strategy of complex networks based on tabu search, Physica A Statistical Mechanics & Its Applications 442 (2016) 74–81. https://doi.org/10.1016/j.physa.2015.08.043.

[18] Q. Li, S.-Y. Liu, X.-S. Yang, Neighborhood information-based probabilistic algorithm for network disintegration, Expert Systems with Applications 139 (2020) 112853. https://doi.org/10.1016/j.eswa.2019.112853.

[19] A. Patron, R. Cohen, D. Li, S. Havlin, Optimal cost for strengthening or destroying a given network, Physical Review E 95 (2017) 052305. https://doi.org/10.1103/PhysRevE.95.052305.

[20] X.-S. Yang, S. Deb, Multiobjective cuckoo search for design optimization, Computers & Operations Research 40 (2013) 1616–1624.

[21] H. Ishibuchi, T. Murata, A multi-objective genetic local search algorithm and its application to flowshop scheduling, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 28 (1998) 392–403.

[22] V. A. Shim, K. C. Tan, H. Tang, Adaptive memetic computing for evolutionary multiobjective optimization, IEEE Transactions on Cybernetics 45 (2014) 610–621.

[23] S. Iyer, T. Killingback, B. Sundaram, Z. Wang, Attack robustness and centrality of complex networks, PLoS One 8 (2013) e59613. http://dx.doi.org/10.1371/journal.pone.0059613.

[24] K. Klemm, M. A. Serrano, V. M. Eguiluz, M. S. Miguel, A measure of individual role in collective dynamics, Scientific Reports 2 (2012) 292–292. http://dx.doi.org/10.1038/srep00292.

[25] J. Liu, J. Lin, Q. Guo, T. Zhou, Locating influential nodes via dynamics-sensitive centrality, Scientific Reports 6 (2016) 21380–21380. http://dx.doi.org/10.1038/srep21380.

[26] A. Barabasi, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512. https://doi.org/10.1515/9781400841356.349.

[27] D. S. Callaway, M. E. Newman, S. H. Strogatz, D. J. Watts, Network robustness and fragility: Percolation on random graphs, Physical review letters 85 (2000) 5468. https://doi.org/10.1103/PhysRevLett.85.5468.

[28] R. Pastorsatorras, Epidemic spreading in scale-free networks, Physical Review Letters 86 (2001) 3200–3203. https://doi.org/10.1515/9781400841356.493.

[29] R. Cohen, K. Erez, D. Ben-Avraham, S. Havlin, Breakdown of the internet under intentional attack, Physical Review Letters 86 (2001) 3682. https://doi.org/10.1103/physrevlett.86.3682.

[30] R. Rabade, N. Mishra, S. Sharma, Survey of influential user identification techniques in online social networks, in: Recent advances in intelligent informatics, Springer, 2014, pp. 359–370. https://doi.org/10.1007/978–3–319–01778–5_37.

[31] P. Holme, B. J. Kim, C. N. Yoon, S. K. Han, Attack vulnerability of complex networks, Physical Review E Statistical Nonlinear & Soft Matter Physics 65 (2002) 056109. https://doi.org/10.1103/PhysRevE.65.056109.

[32] Brandes, Ulrik, A faster algorithm for betweenness centrality*, Journal of Mathematical Sociology 25 (2001) 163–177. https://doi.org/10.1080/0022250X.2001.9990249.

[33] M. E. J. Newman, Clustering and preferential attachment in growing networks, Phys Rev E Stat Nonlin Soft Matter Phys 64 (2001) 025102. https://doi.org/10.1103/PhysRevE.64.025102.

[34] S. P. Borgatti, Centrality and network flow, Social networks 27 (2005) 55–71. https://doi.org/10.1016/j.socnet.2004.11.008.

[35] L. C. Freeman, A set of measures of centrality based on betweenness., Sociometry 40 (1977) 35–41. https://doi.org/10.2307/3033543.

[36] D. Anggraini, S. Madenda, E. P. Wibowo, L. Boumedjout, Network disintegration in criminal network, in: 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, pp. 192–199. https://doi.org/10.1109/SITIS.2015.27.

[37] Y. Deng, J. Wu, Optimal attack strategy based on limited cost model on complex network, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, pp. 105–108. https://doi.org/10.1109/SMC.2015.31.

[38] Y. Deng, J. Wu, Y. Xiao, Y. Li, Efficient disintegration strategies with cost constraint in complex networks: The crucial role of nodes near average degree, Chaos: An Interdisciplinary Journal of Nonlinear Science 28 (2018) 061101. https://doi.org/10.1063/1.5029984.

[39] A. X. Valente, A. Sarkar, H. A. Stone, Two-peak and three-peak optimal complex networks., Physical Review Letters 92 (2004) 118702. https://doi.org/10.1103/physrevlett.92.118702.

[40] R. Albert, H. Jeong, A. L. Barabasi, Error and attack tolerance of complex networks, Nature 340 (2000) 378–382. https://doi.org/10.1515/9781400841356.503.

[41] B. Amiri, L. Hossain, J. W. Crawford, R. T. Wigand, Community detection in complex networks: Multi-objective enhanced firefly algorithm, Knowledge-Based Systems 46 (2013) 1–11. http://dx.doi.org/10.1016/j.knosys.2013.01.004.

[42] Z. Li, J. Liu, K. Wu, A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks, IEEE Trans. Cybern. 48 (Jul. 2017) 1963–1976. http://dx.doi.org/10.1109/TCYB.2017.2720180.

[43] X. Zhang, K. Zhou, H. Pan, L. Zhang, X. Zeng, Y. Jin, A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks, IEEE Trans. Cybern. 50 (Feb. 2020) 703–716. http://dx.doi.org/10.1109/TCYB.2018.2871673.

[44] Q. Li, S.-Y. Liu, X.-S. Yang, Influence of initialization on the performance of metaheuristic optimizers, Applied Soft Computing Journal (2020) https://doi.org/10.1016/j.asoc.2020.106193.

[45] M. G. Everett, S. P. Borgatti, Extending centrality, Models and methods in social network analysis 35 (2005) 57–76. https://doi.org/10.1145/2487788.2488173.

[46] C. A. C. Coello, An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), volume 1, pp. 3–13. http://dx.doi.org/10.1109/CEC.1999.781901.

[47] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE transactions on evolutionary computation 6 (2002) 182–197. https://doi.org/10.1109/4235.996017.

[48] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (2007) 712–731. http://dx.doi.org/10.1109/TEVC.2007.892759.

[49] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (1977) 452–473. https://doi.org/10.1086/jar.33.4.3629752.

[50] J. Kunegis, Konect: the koblenz network collection, in: Proceedings of the 22nd international conference on world wide web, pp. 1343–1350. https://doi.org/10.1145/2487788.2488173.

[51] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, S. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (2003) 396–405. http://dx.doi.org/10.1007/s00265–003–0651–y.

[52] S. Srinivas, C. Rajendran, Community detection and influential node identification in complex networks using mathematical programming, Expert Systems With Applications 135 (2019) 296–312. http://dx.doi.org/10.1016/j.eswa.2019.05.059.

[53] P. M. Gleiser, L. Danon, Community structure in jazz, Advances in Complex Systems 06 (2003) 565–573. http://dx.doi.org/10.1142/S0219525903001067.

[54] S. Mugisha, H. Zhou, Identifying optimal targets of network attack by belief propagation, Physical Review E 94 (2016) 012305–012305.

[55] P. Erdos, A. Renyi, On the evolution of random graphs, Transactions of the American Mathematical Society 286 (2012) 257–274. https://doi.org/10.1515/9781400841356.38.

[56] D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998) 440–442. https://doi.org/10.1515/9781400841356.301.

[57] X.-S. Yang, Nature-inspired optimization algorithms, Academic Press, 2020.

[58] A. Ferligoj, A. Kramberger, Recall versus recognition: comparison of the two alternative procedures for collecting social network data, in: Develop. in Stat. and Methodology: Proc. of the Intl. Conf. on Methodology and Stat., Bled, Slovenia, volume 9, p. 121.