Robustness Optimization Scheme With Multi-Population Co-Evolution for Scale-Free Wireless Sensor Networks

Tie Qiu, *Senior Member, IEEE, ACM*, Jie Liu, Weisheng Si, *Member, IEEE*, and Dapeng Oliver Wu, *Fellow, IEEE*

***Abstract*— Wireless sensor networks (WSNs) have been the popular targets for cyberattacks these days. One type of network topology for WSNs, the scale-free topology, can effectively withstand random attacks in which the nodes in the topology are randomly selected as targets. However, it is fragile to malicious attacks in which the nodes with high node degrees are selected as targets. Thus, how to improve the robustness of the scale-free topology against malicious attacks becomes a critical issue. To tackle this problem, this paper proposes a Robustness Optimization scheme with multi-population Co-evolution for scale-free wireless sensor networKS (ROCKS) to improve the robustness of the scale-free topology. We build initial scale-free topologies according to the characteristics of WSNs in the real-world environment. Then, we apply our ROCKS with novel crossover operator and mutation operator to optimize the robustness of the scale-free topologies constructed for WSNs. For a scale-free WSNs topology, our proposed algorithm keeps the initial degree of each node unchanged such that the optimized topology remains scale-free. Based on a well-known metric for the robustness against malicious attacks, our experiment results show that ROCKS roughly doubles the robustness of initial scale-free WSNs, and outperforms two existing algorithms by about 16% when the network size is large.**

***Index Terms*— Wireless sensor networks, robustness, scale-free topology, multi-population co-evolution.**

I. Introduction

WIRELESS sensor networks (WSNs) [1]–[3] have become a hot research field with a broad range of applications. Typically, WSNs deploy a large number of network nodes within a certain area, and these nodes communicate with each other to monitor or control environmental parameters such as temperature, lighting, etc. Due to the prevalence of cyber-attacks, how to improve the robustness of WSNs becomes an essential issue in recent years [4].

Network topology abstracts how nodes are connected in a network. It is the foundation for communication activities happening inside the network. Complex network theory [5] studies the network topologies of an important class of networks called Complex Networks in which the topological feature is neither purely regular (e.g., in a lattice graph [6]) nor purely random (e.g., in a random graph [7]). Actually, many kinds of real-world networks such as the Internet, social networks, brain networks, etc. are all complex networks.

There are two classic models in complex network theory, one is small world topology and the other is scale-free topology. The small world model has two notable features [8], [9], which are small average path length and high clustering coefficient [10]. It is generally used in modeling heterogeneous network topologies in WSNs [11]. On the other hand, scale-free model is characterized by the power-law distribution of node degrees, and mainly used in modeling homogeneous network topologies [12], [13]. Since the scale-free model has a power-law distribution of node degrees, it is robust against random attacks, but vulnerable to malicious attacks [14]. Therefore, researchers have been focusing on how to enhance scale-free topologies to withstand malicious attacks.

Some proposed approaches try to enhance the robustness of networks with Genetic Algorithm(GA) [15]. Due to single population of candidate solutions, it brings a typical limitation called premature convergence [16], in which the evolution falls into a local optimum too early, resulting in a solution far from the global optimum. Besides, it is known that multi-population genetic algorithm can effectively overcome this limitation by using multiple populations to evolve together. Different probability of crossover operator and mutation operator are assigned to each population of multiple-population. Individuals with high fitness values can be introduced into other different populations through migration operator, which can effectively prevent falling into a local optimum. Therefore, in this paper, we propose to use multi-population co-evolution to enhance the robustness of scale-free topologies. To validate this idea, we give a concrete scheme called Robustness Optimization with multi-population Co-evolution for scale-free

Manuscript received November 20, 2017; revised July 16, 2018 and January 3, 2019; accepted March 19, 2019; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Shin. Date of publication April 12, 2019; date of current version June 14, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61672131. The work of W. Si was supported in part by a grant from NSW Cyber Security Network. *(Corresponding authors: Tie Qiu; Dapeng Oliver Wu.)*

T. Qiu is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Advanced Networking, Tianjin 300350, China (e-mail: qiutie@ieee.org).

J. Liu is with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: liujie.dut@gmail.com).

W. Si is with the School of Computing, Engineering and Mathematics, Western Sydney University, Parramatta, NSW 2150, Australia (e-mail: w.si@westernsydney.edu.au).

D. O. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: wu@ece.ufl.edu).

Digital Object Identifier 10.1109/TNET.2019.2907243

wireless sensor networKS (ROCKS). ROCKS introduces novel crossover and mutation operators to rewire the edges in network topologies to enhance the robustness against malicious attacks.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 gives an overview of ROCKS. Section 4 details the crossover operator and mutation operator in ROCKS by giving their algorithms. Section 5 presents simulation results on ROCKS and compare it with two existing algorithms. Finally, Section 6 concludes this paper.

II. Related Work

The scale-free network has the typical characteristics that most nodes (ordinary nodes) have a small node degree, while very few nodes (hub nodes) have very large node degrees [12], [13]. This characteristic makes scale-free networks robust against random attacks (mostly occurring on ordinary nodes), but fragile under malicious attacks (occurring on hub nodes). This paper focuses on how to improve the robustness of scale-free wireless sensor networks against malicious attacks. Although increasing the number of links among the nodes could enhance the robustness of scale-free networks, it will destroy the scale-free characteristic of the network. Besides, Increasing the number of links also increases the energy consumption of the network. Therefore, we explore an effective way to improve the robustness of scale-free WSNs against malicious attacks, during which the initial degree of each node keeps unchanged such that the optimized topology remains scale-free [17].

Due to low node power and energy saving consideration, some researchers are working to reduce energy consumption and energy efficiency [18], [19]. WSNs has the following two constraints: (1) the communication range of its nodes cannot be arbitrarily long and (2) its node degrees cannot be arbitrarily large. Because of these two constraints, the traditional method for constructing scale-free topologies, the Barabási Albert model (BA model) [20], cannot be directly applied. So, we apply the following method, which adapts the BA model, to construct scale-free topologies for WSNs.

We add edges between nodes sequentially during the process of constructing WSNs topology after judging whether they are within the communication range. Here 'sequentially' means that a pair of nodes cannot generate a new edge at the same time. The local world of the newly joined node is composed of the all the nodes within its communication range. If a node has been connected with the newly joined node or it has reached the maximum degree, it will be removed from the local world of the newly joined node. Furthermore, the newly joined node chooses neighbors to establish a connection by roulette method according to the degree of nodes. And the newly joined node prefers to connect with higher degree node in its local world.

In recent years, some researchers try to use the GA to solve the problems encountered in the deployment of wireless sensor networks. Shukla *et al.* [21] presented a GA-based routing scheme, which established a trade-off between energy efficiency and energy balancing. Elhoseny *et al.* [22] proposed a self-clustering method for the heterogeneous networks using GA that optimizes the network lifetime. Peiravi *et al.* [23] gave a new GA-based clustering algorithm to simultaneously optimize network lifetime and delay. Zhou and Liu [24] proposed a new memetic algorithm to enhance the robustness of scale-free network against malicious attacks, during which the initial degree of each node kept unchanged. Especially, it is a type of significant enhancing method combine both global and local searching. But it does not consider the limitation of communication range for WSNs nodes. In this paper, by using the evolutionary optimization of GA, we try to explore the best topology scheme for scale-free WSNs, during which the initial degree of each node kept unchanged.

Besides, some researchers became interested in how to effectively optimize the robustness of the scale-free network. Based on percolation theory, Schneider *et al.* [25] proposed a new metric of robustness. They considered the largest connected subgraph [26] when one repeatedly removes the highest-degree nodes in the network to weight the network robustness. Buesser *et al.* [27] used probabilistic swapping strategy to deal with the multimodal phenomenon and enhances the robustness of scale-free network topology, named Simulated Annealing algorithm. This algorithm also can be used in the robustness of scale-free WSNs. But its efficiency is greatly decreased because of redundant swapping edges. Through determining the edges which need to be compared by twice selections, Louzada *et al.* [28] gave a Smart Rewiring method. But it does not consider the limitation of communication range for WSNs nodes, thus, it is not aimed for WSNs.

By using the new metric of robustness mentioned above, Herrmann *et al.* [29] proposed a new algorithm named Hill Climbing, which makes the network topology resemble a stable onion-like structure through swapping edges. But it has multimodal phenomenon which may cause the algorithm jumping into a local optimum. Herrmann *et al.* also have found that onion-like structure is more stable and robust against malicious attacks. Thus, in previous work, we make the evolution of individual topology towards the onion-like structure in mutation operator to improve the robustness of topologies against malicious attacks [30]. Basically, the connections among nodes in onion-like structure exhibit the following characteristics:

- Nodes with similar node degrees connect to each other.
- Node degrees gradually decrease from inner nodes to outer nodes.
- The majorities of the nodes have small degrees and are located in the outer layers of the onion-like structure.

III. Overview of ROCKS

GA is a type of optimization algorithms that imitate the behavior of natural selection in the biological world. It is an iterative process of evolution involving a large number of generations. A GA typically consists of the following components:

- A population of individuals: each individual is a solution to the optimization problem; usually encoded as a 0-1 string.

- A fitness function: it typically calculates the metric for optimization on each individual; used to rank individuals for selection purpose.
- Crossover operator: in this operator, a pair of parents is selected to breed children in the next generation. This operator is typically designed to enable the children to inherit the strengths from their parents and also exhibit diversity.
- Mutation operator: in this operator, a child in the new generation is typically changed in certain stochastic way to increase the diversity of the new generation.

Crossover operators in GA is used to increase the genetic diversity of the population by selecting different individuals as parent samples to generate new offsprings. Mutation operator is used to generate children with high fitness function values directly. With the widespread application of conventional GA, many defects are exposed. One of the fatal issues is the premature convergence mentioned in the previous section.

To address the premature convergence issue, the multi-population genetic algorithm in ROCKS brings the following improvements:

The framework of the GA which only uses single population to evolve is broken by using several populations to search the optimal solution. For different populations, we adopt different crossover probability P_{cro} and mutation probability P_{mut}. Although the ranges of crossover probability and mutation probability are suggested to set as P_{cro}(0.7-0.9) and P_{mut}(0.001-0.05) respectively, their ranges are wide enough to allow plenty of probability values being experimented. The optimization results will differ greatly with different P_{cro}'s and P_{mut}'s. Given the global search and local search at the same time, we conduct experiments with various P_{cro}'s and P_{mut}'s to pick the best values of them to prevent different populations from falling into the local optimum.

Each population is independent and communicates with each other through migration operator. migration operator will move the optimal individual that appeared on each population during evolution to other populations in each generation, which achieves the gene exchange among populations.

The best individual that appears in each generation of evolution is selected to compose the migration population. We do not have the operation of crossover and mutation in migration population, which ensures the best individual of each population will not be destroyed. The migration population is the foundation of migrant operator, which increases the genetic diversity, and guarantees the fitness function to search for the optimal solution in a wide range.

For given scale-free topologies for WSNs, we optimize their robustness using ROCKS. Firstly, we convert the adjacency matrix of topology to a binary-coded chromosome. To further illustrate this operator, a topology with five nodes is converted to a chromosome in Fig. 1. The topology consists of node i, node j, node k, node l and node m, and the adjacency matrix is a binary matrix. It is feasible to convert the adjacency matrix into a chromosome directly. However, the storage space is wasted and the operating complexity is increased in the GA when dealing with a huge network. The adjacency matrix is a symmetric matrix, and its upper triangular matrix is able to completely represent the connections between nodes in the network. We convert the upper triangular matrix to a chromosome as shown in Fig. 1. It can shorten the length of the chromosome and improve the efficiency of the GA.

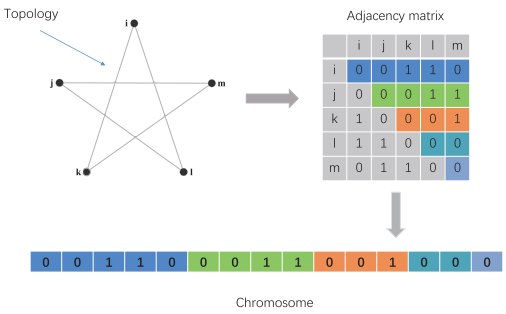


Fig. 1. The adjacency matrix is converted to a chromosome.

Each individual will be evaluated in each generation of the GA. We get the fitness value of individuals by calculating the fitness function. Individuals of each population will be sorted by the fitness value. Individuals with high fitness values are more likely to be selected to enter the next generation. The fitness function directly influences the evolution direction of the GA. Thus, it is very important to construct an appropriate fitness function for the ROCKS proposed in this paper.

In order to measure the robustness of the scale-free network against malicious attacks, we use the attack strategy called High Degree Adaptive (HDA) [31]. In HDA, all nodes in the network are sorted according to their node degrees. In each round of attack, the node with the highest degree and the edges connected to this node are removed.

To evaluate the network robustness under the HDA attack, Schneider *et al.* [25] proposed a metric called R, which considers the maximal connected subgraph after each round of attack. Specifically, R is defined in Eq. (1) below:

$$\mathrm{R} = \frac{1}{N+1}\sum_{n=0}^{N}\frac{MCS(n)}{N} \tag{1}$$

Wherein, N represents the total number of nodes in the network; n represents the nth round of attack; $MCS(n)$ represents the number of nodes in the largest connected subgraph after the nth round of attack; the summation considers N round of attacks until all nodes are removed; and the normalization factor $1/(N+1)$ ensures that the networks with different sizes and edge densities can be compared. Note that the value of R lies in the range (0, 0.5] [29].

It is obvious that the higher the value of R is, the higher the robustness of scale-free WSNs topology. Thus, we employ metric R as the fitness function $f(G)$ in ROCKS to measure the robustness of WSNs topologies, which will guide all the populations evolving to resist malicious attacks.

IV. ROCKS

A. Initialization Operation

Individual diversity at the initialization operation is the basis of genetic diversity in evolutionary processe of ROCKS. If the difference among individuals is little, the advantage

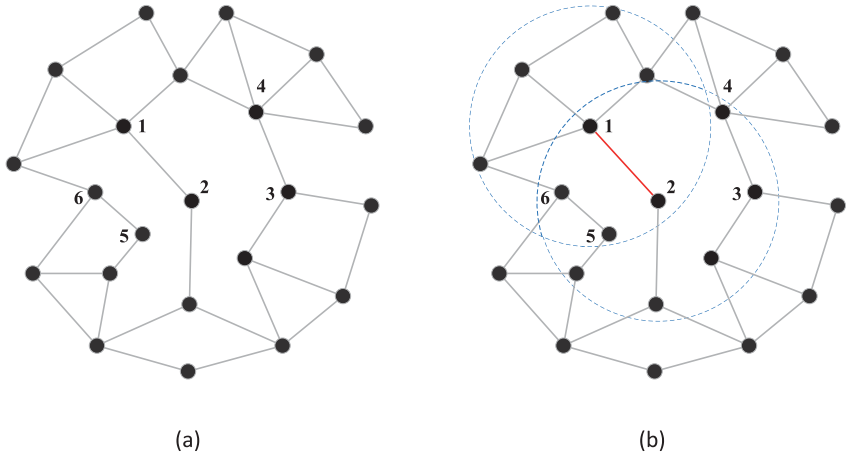


Fig. 2. Initial topology and how to select edges for swapping. (a) Initial topology. (b) Select edges for swapping.

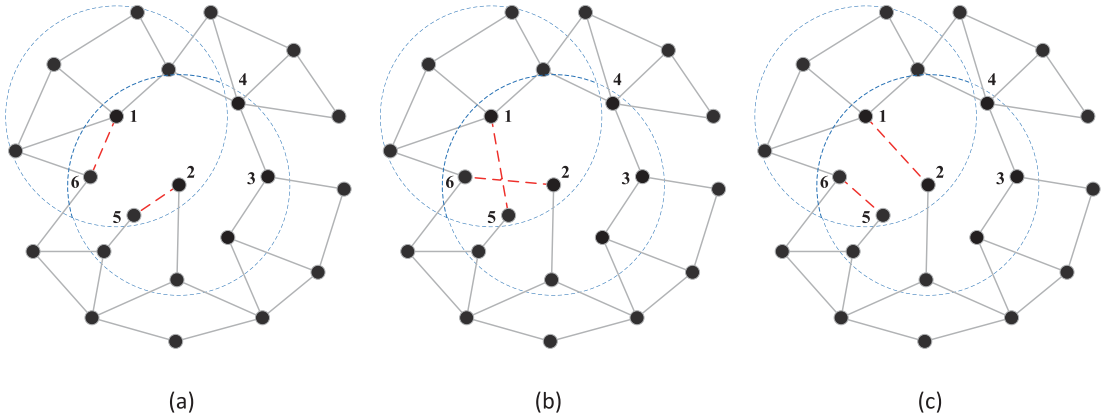


Fig. 3. Three edge-swapping methods. (a) Method1. (b) Method2. (c) Method3.

of crossover operator will disappear soon. And the evolution of species will depend on the mutation operator only. Since the probability of mutation operator is small, the evolution of species will be close to a single state soon, which causes the termination of evolution. In order to ensure that the individuals among population have a big difference in the initial time, we design the following initialization operation.

For an original topology of scale-free WSNs, the location of every node is stationary. We make the following transformation under the premise that the degree distribution does not change. Firstly, we assign a random probability between [0,1] as P_{init} for each individual of each population. P_{init} controls the frequency of edge-swapping on the original topology. Then for the selected edges, we select one of the following three swapping methods in Fig. 3. Thus, each individual transformed from the original topology is completely random and has a large difference from others.

Fig. 2(a) is an original scale-free WSNs topology, and we will do the initialization algorithm on it to generate a new topology which is different from the original topology but has the same degree distribution. Firstly, a random number between [0,1] is generated as the edge-swapping frequency P_{init}. Then each edge of the topology is traversed one by one. For each edge, P_{init} is compared with a random probability rv between [0,1]. If rv is more than P_{init}, we will proceed to the next edge. If rv is less than P_{init}, we will do the edge-swapping operation. As shown in Fig. 2(b), We have selected edge e_{12} between node 1 and node 2 for edge swapping operation. Then we traverse each edge of the topology that is not adjacent to e_{12}, and look for the object of edge-swapping operation for e_{12}. For e_{34} in Fig. 2(b), both node 3 and note 4 are in the communication radius of node 2, but they are not in the communication radius of node 1. Thus, e_{34} is discarded. In the process of traversal, we find a suitable edge e_{56}, whose node 5 and node 6 are in the intersection of node 1's and node 2's communication ranges. Finally, we determine e_{12} and e_{56} for edge-swapping operation. As shown in Fig. 3, (a), (b) and (c) correspond to three candidate operations respectively. We randomly select one of them to conduct edge-swapping. Fig. 3(c) means there is no operation on the two selected edges. If a matching object for e_{12} cannot be found after traversing all the edges of topology, the edge-swapping operation for e_{12} will be aborted, and we will traverse the remaining edges.

B. Crossover Operator

The optimization ability of genetic algorithm comes from population genetic diversity. Crossover operator is used to

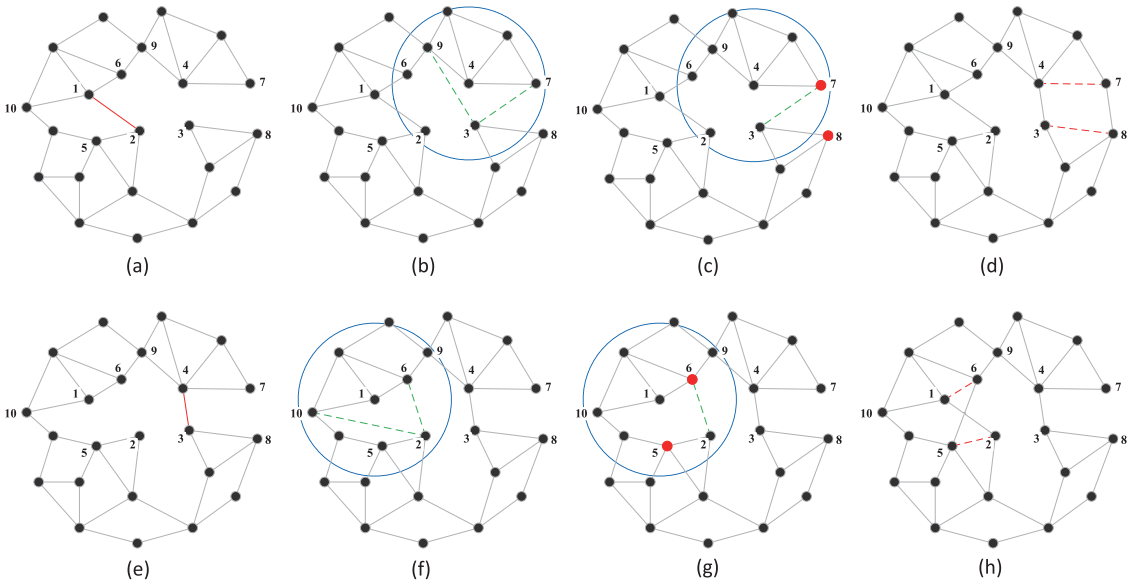


Fig. 4. The process of crossover operator. (a) Step1. (b) Step2. (c) Step3. (d) Step4. (e) Step5. (f) Step6. (g) Step7. (h) Step8.

increase the genetic diversity of the population by selecting different individuals as parent samples to generate new offsprings. It has no fixed evolution direction in the ROCKS. Parent topologies generate new children topologies by crossover operator, which obtains a larger solution space. Thus, the fitness function will search the best solution in a larger space. Generally, crossover operator retains part of father and mother genes, and eventually generates new children topologies. The crossover operator in this paper keeps the initial degree of each node unchanged, which means that we will preserve the degree distribution of parent topologies.

Taking into account the communication range in WSNs, we design the crossover operator as follows.

Suppose G_f and G_m are the father's topology and mother's topology, respectively. And G_s and G_d are the son's topology and daughter's topology, respectively. Firstly, the parents are chosen by the probability P_{cro} of crossover operator in a population. Then G_s inherits its father's topology G_f and G_d inherits its mother's topology G_m. Obtain the following sets of edges:

$$E_f^G = \{e_{ij} | e_{ij} \in G_f\} \tag{2}$$

$$E_m^G = \{e_{ij} | e_{ij} \in G_m\} \tag{3}$$

$$E_f = E_f^G - (E_f^G \cap E_m^G) \tag{4}$$

$$E_m = E_m^G - (E_m^G \cap E_f^G) \tag{5}$$

where, E_f^G is the set of the father's edges, and E_m^G is the set of mother's edges. E_f and E_m are the sets of father's exclusive edges and mother's exclusive edges respectively. Here 'exclusive' means an edge only exists in one parent's set but not the other. That is, E_f is totally different from E_m. Because the location of each node is stationary, if one edge exists in E_f, we also can build it in G_d. Finally, the son's topology disconnects the existing edges to build every mother's exclusive edges E_m, during which the initial degree of each node is kept unchanged. And the construction process of daughter's topology is similar to above operation. Fig. 4 illustrates the process of crossover operator.

Fig. 4(a) and Fig. 4(e) represent the connections between nodes in the father's topology G_f and mother's topology G_m. It can be seen that the father has an exclusive edge e_{12} between node 1 with node 2 in Fig. 4(a), and mother has an exclusive edge e_{34} between node 3 with node 4 in Fig. 4(e). According to the criteria in crossover operator, we build the mother's exclusive edge e_{34} in the son's topology (Fig. 4(d)), and the father's exclusive edge e_{12} in daughter's topology (Fig. 4(h)).

Here is the detailed description about how father's topology (Fig. 4(a)) generates his son's topology (Fig. 4(d)). In order to generate a new edge e_{34} in Fig. 4(a), we select the candidate nodes which have no edge with the node 3 in the neighbors of the node 4. Then we calculate the distance of the candidate nodes to the node 3. Finally, we sort the distances to generate a candidate list in ascending order. As shown in Fig. 4(b), the node 7 that is the neighbor of the node 4 and has no edge with the node 3, is the nearest node to node 3. Node 3 searches each of its neighbor nodes in Fig. 4(c) until finds a node which is in the communication range of the node 7 and has no edge with the node 7. As shown in Fig. 4(c), the node 3 chooses its neighbor node 8, and we disconnect the edges e_{47} and e_{38} in Fig. 4(d). After that, we generate the edges e_{34} and e_{78}. Finally, we successfully generate a new edge e_{34} in son's topology (Fig. 4(d)). The degree of node 3 and node 4 both equal to 3 before crossover operator, and after the operator they still keep unchanged. Therefore, it is consistent with the criterion that keeps the initial degree of each node unchanged. The process that the mother generates its daughters topology is similar to the above operations as shown in Fig. 4(e-g). Finally, we can see the father's exclusive edge e_{12} in the daughter's topology in Fig. 4(h).

Besides, when the father's topology generates his son's topology, if the node 3 cannot find an eligible node to match the node 7 which is the candidate neighbor of the node 4,

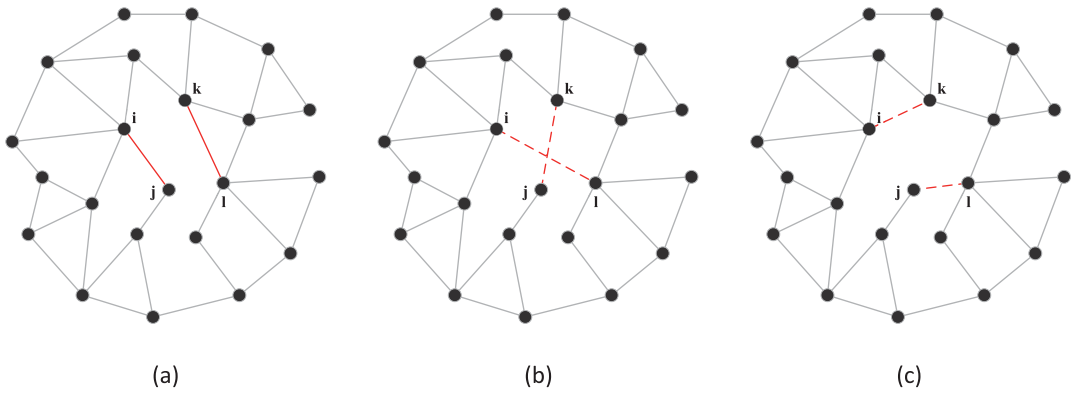


Fig. 5. Candidates for the topology connection. (a) Select edges e_{ij} and e_{kl}. (b) Candidate1. (c) Candidate2.

the node 4 will sequentially choose another candidate node in the candidate list. And the node 3 will search all of its neighbors for each candidate node until it finds an eligible node to match the candidate node. If the node 3 still cannot find an eligible node after traversing the candidate list of the node 4, we give up generating this edge.

C. Mutation Operator

The mutation operator in ROCKS not only increases the diversity of a new generation but also produces individuals with high fitness values. We choose the individual by the mutation probability P_{mut}. The goal of mutation operator is to increase the robustness of the selected individual through exchanging edges, during which the initial degree of each node is unchanged. Metric R is used to measure the robustness of topology. We search for optimal solution within the local area by the mutation operator.

Herrmann *et al.* [29] have found that onion-like structure is stable and robust against malicious attacks. The nodes with similar degrees connect to each other in onion-like structure. If a node with a large degree failed, another node with a large degree will replace its function. Therefore, we can minimize the adverse effects of failure nodes as much as possible, and the network topology will remain robust. In order to make the evolution of individual topology toward the onion-like structure, we generate a new edge between two nodes that have a similar degree, during which the initial degree of each node is unchanged. We select two edges in the individual topology and judge the four end nodes of these two edges whether they are in the communication range of each other to guarantee that we can generate a new edge among these four nodes.

For the edges e_{ij} and e_{kl} selected in Fig. 5(a), we propose a criterion to sort degree and swap edges as follows Eq. (6).

$$\frac{d_1 - d_2 + d_3 - d_4}{|d_i - d_j| + |d_k - d_l|} < P_{swap} \tag{6}$$

Wherein, d_i, d_j, d_k, d_l are the degree of node i, node j, node k and node l respectively. We sort them in descending order, and name them as d_1, d_2, d_3, d_4. P_{swap} controls reduction ratio of degree difference. If the formula on the left is less than P_{swap}, we will swap edges according to d_1, d_2, d_3, d_4. There are two candidate strategies in Fig. 5(b) and Fig. 5(c).

Based on the criteria mentioned above, the nodes that have similar degrees will connect with each other, thus enabling the evolution of individual topology towards the onion-like structure. Besides, the swapping threshold P_{swap} is defined in [0, 1), and it cannot be 1 because the two edges will not be swapped in that case. We control the efficiency of mutation operator by adjusting the value of P_{swap}. The appropriate swapping threshold P_{swap} can effectively avoid inefficient swapping edges operation.

D. Migration Operator

The migration operator in ROCKS is designed to overcome premature convergence. Individuals with high fitness values can be introduced into other populations through migration operator, which can effectively prevent falling into a local optimum. We assign different mutation operator and crossover operator probabilities for different populations, and the suitable individuals are selected in different populations in every generation. Thus, genetic communication can be carried out between different populations to prevent trapping into local optima.

The migration operation is divided into three steps. Firstly, we select the individual with the highest fitness value in each population. This optimal individual will be temporarily stored in an elite population. Secondly, the worst individual in each population is selected. Finally, each optimal individual stored in the elite population will be used to replace a worst individual in a different population.

Fig. 6 depicts a complete process of migration operation. Each population has ten individuals. And each column represents an individual, and its length indicates the value of the individual fitness function. The longer the column is, the larger the value of individual fitness function will be. We identify the best individual of each population and mark it by bright color in Fig. 6(a). The selected individuals make up the elite population in Fig. 6(b). Then as shown in Fig. 6(b), the worst individual in each population is identified and marked by red. Finally, in Fig. 6(c), the current optimal individual of population 1 is used to replace the worst individual of population 2. The current optimal individual of population 2 is used to replace the worst individual of population 3. And the current

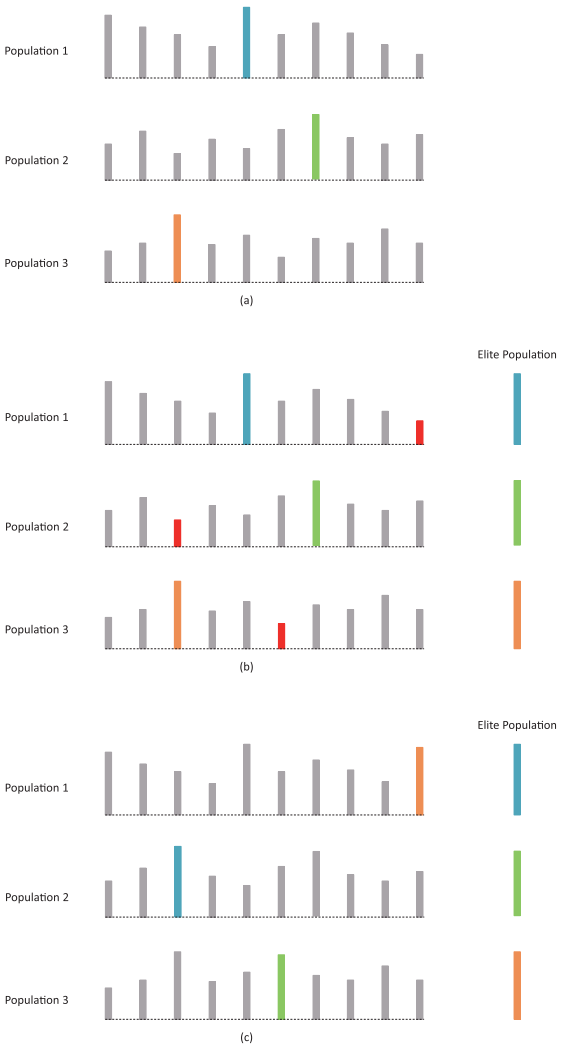


Fig. 6. Migration operator. (a) Select the optimal individual. (b) Identify the worst individual. (c) Replacement.

optimal individual of population 3 is used to replace the worst individual of population 1.

E. Algorithm Design

In this subsection, we detail the algorithms for the four operations described in the previous subsections.

1) Initialization Algorithm: Algorithm 1 describes the detailed process of initialization operation. The variables used in the algorithm are as follows.

- G_0: Original topology of a scale-free WSNs.
- M_{pop}: Size of population.
- N_{ind}: Number of individuals.
- G^{init}: The initialization matrix of the ROCKS.
- $G_j^{init_i}$: The element of ith row and jth column of matrix G^{init}.
- E_0: The sets of all edges in the initial network.
- E_n: The sets of all edges in the current network.
- rv: A random variable that uniformly distributed in [0,1].
- $ComRang_{ab}$: The intersection communication rang of node a and node b.
- e_{ab}^{Nei}: The neighbor of e_{ab}.

Algorithm 1 ROCKS Initialization

Input: G_0, M_{pop}, N_{ind}.
Output: G^{init}

1: **for** $i = 1 : M_{pop}$ **do**
2: **for** $j = 1 : N_{ind}$ **do**
3: $G_j^{init_i} \leftarrow G_0$
4: $E_n \leftarrow E_0$
5: $P_{init} = rand(0, 1)$
6: **for all** $e_{ab} \in E_n$ **do**
7: $E_n = E_n - \{e_{ab}\}$
8: $rv \leftarrow rand(0, 1)$
9: **if** $rv < P_{init}$ **then**
10: Search in E_n until find an edge e_{cd} that:
11: $e_{cd} \in ComRang_{ab}$ and $e_{cd} \notin e_{ab}^{Nei}$
12: $candidate \leftarrow$ A random number in $[1, 2, 3]$;
13: **if** $(candidate == 1)$ **then**
14: Remove $\{e_{ab}, e_{cd}\}$ from $G_j^{init_i}$
15: Add $\{e_{ac}, e_{bd}\}$ to $G_j^{init_i}$
16: **else**
17: **if** $(candidate == 2)$ **then**
18: Remove $\{e_{ab}, e_{cd}\}$ from $G_j^{init_i}$ A
19: Add $\{e_{ad}, e_{bc}\}$ to $G_j^{init_i}$
20: **else**
21: No operation
22: **end if**
23: **end if**
24: $E_n = E_n - \{e_{cd}\}$
25: **if** The edge does not exist **then**
26: $Continue$
27: **end if**
28: **end if**
29: **end for**
30: **end for**
31: **end for**

- P_{init}: The edge-swapping frequency.
- e_{ab}: The edge selected currently.
- e_{cd}: The edge-swapping object of e_{ab}.

This algorithm works as follows. For the original topology, a random probability between [0,1] is generated for every initial topology, which is the P_{init} (Lines 3 - 5). We traverse each edge of the initial topology (Line 6). A random number rv is generated to compare with the edge-swapping frequency P_{init} (Line 8). If rv is less than P_{init}, we enter the edge-swapping operation. Firstly, we find a target edge that can be swapped with the edge selected currently in set E_n. The target edge should satisfy the conditions in line 10. Then the edge-swapping scheme is selected in random (Line 13). Line 12 to line 22 is the specific of edge-swapping operation. If the target cannot be found, the edge-swapping operation for the current edge will be aborted (Line 24 - 26). Finally, each individual of each initial population is assigned a different topology from the original topology, but has the same degree distribution with the original topology.

2) Crossover Algorithm: Algorithm 2 describes the detailed process of crossover operator, which is an extremely important

Algorithm 2 Crossover Operator

Input: G^i, N_{ind}, P_{cro}.
Output: G^{cro_i}

1: $inum \leftarrow 0$
2: $G^{cro_i} \leftarrow G^i$
3: **for** $j = 1 : N_{ind}$ **do**
4: $rv \leftarrow rand(0, 1)$
5: **if** $rv < P_{cro}$ **then**
6: $reg_{inum} \leftarrow j$
7: $inum++$
8: **end if**
9: **if** $(inum == 2)$ **then**
10: $G_f \leftarrow G^i_{reg_1}$
11: $G_m \leftarrow G^i_{reg_2}$
12: $E_f = E^G_f - (E^G_f \cap E^G_m)$
13: $E_m = E^G_m - (E^G_m \cap E^G_f)$
14: $G_s \leftarrow G_f$
15: $G_d \leftarrow G_m$
16: /*Build every exclusive edge of father's topology in daughter's topology*/
17: **for all** $e_{ij} \in E_f$ **do**
18: $E_f = E_f - \{e_{ij}\}$
19: **for all** node $k \in Lcand$ **do**
20: Select in Lst_j until find node l that:
21: node $l \in ComRang_k$ and $e_{lk} \notin E_f$
22: **if** $node\ l\ exists$ **then**
23: Remove $\{e_{ik}, e_{jl}\}$ from G_d
24: Add $\{e_{ij}, e_{kl}\}$ to G_d
25: $Break$
26: **end if**
27: **end for**
28: **if** $node\ l\ does\ not\ exist$ **then**
29: $Continue$
30: **end if**
31: **end for**
32: /*Build every exclusive edge of mother's topology in son's topology as above*/
33: $G^i_{reg_1} \leftarrow G_s$
34: $G^i_{creg_2} \leftarrow G_d$
35: $inum \leftarrow 0$
36: **end if**
37: **end for**

part of ROCKS. The variables used in the algorithm are as follows.

- G^i: The set of individuals in the ith population.
- P_{cro}: The probability of crossover operator.
- G^{cro_i}: The ith population after the crossover operator.
- reg: The index of individual that participates in crossover operator.
- reg_{inum}: The $inum$th element in register reg.
- $G^i_{reg_1}$: Topology of the reg_1th individual of the ith population.
- G_f: The father's topology in crossover operator.
- G_m: The mother's topology in crossover operator.
- G_s: The son's topology in crossover operator.
- G_d: The daughter's topology in crossover operator.
- E^G_f: The sets of edges in the father's topology.
- E^G_m: The sets of edges in the mother's topology.
- E_f: The sets of father's exclusive edges.
- E_m: The sets of mother's exclusive edges.
- e_{ik}: The edge selected currently.
- e_{jl}: The edge-swapping object of e_{ik}.
- Lst_i: The list of neighbors connected to node i.
- Lst_j: The list of neighbors connected to node j.
- $ComRang_k$: The communication rang of node k
- $Lcand$: The list of candidate node that ascending sort the distances of node in Lst_i to node j.

Every individual of the input population will be traversed (Line 3). A random number rv between [0,1] is generated to compare with the probability of crossover operator P_{cro} (Line 4). If rv is less than P_{cro}, the current individual will be selected as one of the parent topology (Lines 5 - 6). After we have a couple of individuals, the crossover operator is beginning (Line 8). One of the couples becomes the father's topology G_f, and the other becomes the mother's topology G_m (Lines 9 - 10). Then we get the set of father's exclusive edges E_f and the set of mother's exclusive edges E_m (Lines 11 - 12). The son's topology inherits its father's topology, and the daughter's topology inherits its mother's topology (Lines 13 - 14). Then we build every exclusive edge of father's in daughter's topology (lines 15 - 30), which has been described in detail in Section 4.2. This process continues until every father's exclusive edge has been rebuilt in daughter's topology. The method of generating the son's topology is similar to the above operations. Finally, the son replaces its father's position and the daughter replaces its mother's position in the population (Lines 32 - 33).

3) Mutation Algorithm: Algorithm 3 describes the detailed process of mutation operator. The variables used in the algorithm are as follows.

- P_{mut}: The probability of mutation operator.
- P_{select}: The frequency of select edges.
- P_{swap}: The threshold of swapping edges.
- G^{mut_i}: The ith population after the mutation operator.
- P_{temp}: The temporary ratio value of degree change.
- $G^{mut_i}_j$: The topology of the jth individual of the ith population in mutation operator.
- G_{mut}: The target topology of crossover operator.
- E^G_{mut}: The sets of edges in the current topology.
- e_{ij}: The edge selected currently.
- e_{kl}: The edge-swapping object of e_{ik}.
- $ComRang_{ij}$: The intersection communication rang of node i and node j.
- e^{Nei}_{ij}: The neighbor of e_{ij}.
- $index$: Records the original index of end nodes in selected edges.
- $R1$: Robustness of the target topology before operation.
- $R2$: Robustness of the target topology after operation.

Every individual of the input population is traversed (Line 2). A random number rv between [0,1] is generated to compare with the probability of mutation operator P_{mut} (Line 3-4). If rv is less than P_{mut}, the current individual

Algorithm 3 Mutation Operator

Input: G^i, N_{ind}, P_{mut}, P_{select}, P_{swap}.
Output: G^{mut_i}

1: $G^{mut_i} \leftarrow G^i$
2: **for** $j = 1 : N_{ind}$ **do**
3: $rv \leftarrow rand(0, 1)$
4: **if** $rv < P_{mut}$ **then**
5: **for all** $e_{ij} \in E^G_{mut}$ **do**
6: $E^G_{mut} = E^G_{mut} - \{e_{ij}\}$
7: $rv2 \leftarrow rand(0, 1)$
8: **if** $rv2 < P_{select}$ **then**
9: Search in E^G_{mut} until find an edge e_{kl} that:
10: $e_{kl} \in ComRang_{ij}$ and $e_{kl} \notin e^{Nei}_{ij}$
11: $P_{temp} = \frac{d_1-d_2+d_3-d_4}{|d_i-d_j|+|d_k-d_l|}$
12: **if** $P_{temp} < P_{swap}$ **then**
13: $G_{mut} \leftarrow G^{mut_i}_j$
14: $R1 \leftarrow CalculateRobust(G^{mut_i}_j)$
15: **if** $(index(1) + index(2) = 5)$ **then**
16: Remove $\{e_{ij}, e_{kl}\}$ from G_{mut}
17: Add $\{e_{il}, e_{jk}\}$ to G_{mut}
18: **else**
19: Remove $\{e_{ij}, e_{kl}\}$ from G_{mut}
20: Add $\{e_{ik}, e_{jl}\}$ to G_{mut}
21: **end if**
22: $R2 \leftarrow CalculateRoust(G_{mut})$
23: **if** $R2 > R1$ **then**
24: $G^{mut_i}_j \leftarrow G_{mut}$
25: **end if**
26: $E^G_{mut} = E^G_{mut} - \{e_{kl}\}$
27: **if** *the edge does not cexist* **then**
28: $Continue$
29: **end if**
30: **end if**
31: **end if**
32: **end for**
33: **end if**
34: **end for**

Algorithm 4 ROCKS

Input: G_0, M_{pop}, N_{ind}, $MaxGen$,
P_{cro}, P_{mut}, P_{select}, P_{swap}.
Output: G_R

1: $G^p \leftarrow Initialization(G_0, M_{pop}, N_{ind})$
2: $P^{cro} \leftarrow P_{cro} + (ConCro - P_{cro}) * rand(M_{pop}, 1)$
3: $P^{mut} \leftarrow P_{mut} + (ConMut - P_{mut}) * rand(M_{pop}, 1)$
4: $genSafe \leftarrow 1$
5: **for** $i = 1 : M_{pop}$ **do**
6: $ObjV^i = ObjectFunction(G^{p_i})$
7: **end for**
8: **while** $(genSafe < MaxGen)$ **do**
9: **for** $i = 1 : M_{pop}$ **do**
10: $G^* \leftarrow G^{p_i}$
11: $G^* = Cros\_OP(G^*, N_{ind}, P^{cro}_i)$
12: $G^* = Mut\_OP(G^*, N_{ind}, P^{mut}_i, P_{select}, P_{swap})$
13: $ObjVSel = ObjectFunction(G^*)$;
14: $[G^{p_i}, ObjV^i] = ReiCh(G^{p_i}, ObjV^i, G^*, ObjVSel)$
15: $maxObjV(genSafe, i) = max(ObjV^i)$
16: **end for**
17: **for** $i = 1 : M_{pop}$ **do**
18: $[MaxSubscript] = MaxPo(G^{p_i}, ObjV^i)$
19: $next_i = i + 1$
20: **if** $next_i > M_{pop}$ **then**
21: $next_i = mod(next_i, M_{pop})$
22: **end if**
23: $[MinSubscript] = MinPo(G^{p_{next_i}}, ObjV^{next_i})$
24: $G^{p_{next_i}}_{MinSubscript} \leftarrow G^{p_i}_{MaxSubscript}$
25: $ObjV^{next_i}_{MinSubscript} \leftarrow ObjV^i_{MaxSubscript}$
26: **end for**
27: $EliteIndul(MaxObjV, MaxChromG, G^p, ObjV)$
28: $genSafe = genSafe + 1$
29: **end while**
30: $G_R \leftarrow MaxChromG$

will be chosen to participate mutation operator. Then every edge of the selected individual will be traversed (Line 5). In order to control the frequency of individual's mutation operations, we generate another random number $rv2$ between [0,1] to compare with the frequency of mutation operator P_{select} (Line 7-8). If $rv2$ is less than P_{select}, we find the target edge for the current edge (Line 9-10). We sort the degrees of node i, node j, node k and node l in descending order, and name them as d_1, d_2,d_3, d_4. Then we get the temporary ratio value P_{temp} of degree change, and compare it with the a threshold of swapping edges P_{swap} (Lines 11). Only when P_{temp} is less than P_{swap}, we execute the edge-swapping operation. In this situation, the scheme of the operation is determined (Lines 13 - 21). The robustness of topology before and after the edge-swapping operation are calculated by $CalculateRoust$ (Lines 14, 22). After the edge-swapping operation, if the metric R opposed to before does not increase, the edge-swapping operation will be canceled (Lines 23 - 25). Besides, if the target edge for the current edge cannot be found, the operation of the current edge also will be canceled. This process continues until every edge of the selected individual has been visited.

4) ROCKS: Algorithm 4 describes the detailed process of ROCKS. The variables used in the algorithm are as follows.

- G_R: The individual topology that has the current best robustness in ROCKS.
- G^p: The sets of all current populations, and each row represents a population.
- P^{cro}: The sets of crossover operator probabilities of all populations.
- P^{cro}_i: The crossover operator probabilities of ith population.
- P^{mut}: The sets of mutation operator probabilities of all populations.
- P^{mut}_i: The mutation operator probabilities of ith population.
- $ConCro$: The benchmark of P^{cro}.
- $ConMut$: The benchmark of P^{mut}.
- $genSafe$: The current generation.

- G^{p_i}: The ith population of ROCKS.
- $G^{p_{next_i}}$: The $next_i$th population of ROCKS.
- G^*: The current population.
- $ObjV$: The sets of fitness values for all populations.
- $ObjV^i$: The sets of fitness values for ith population.
- $ObjVSel$: The sets of fitness values of current operating population.
- $MaxSubscript$: The index of the best individual in the current population.
- $MinSubscript$: The index of the worst individual in the current population.
- $G^{p_i}_{MaxSubscript}$: The best individual topology in ith population.
- $G^{p_{next_i}}_{MinSubscript}$: The worst individual topology in $next_i$th population.
- $ObjV^i_{MaxSubscript}$: The fitness value of the best individual in ith population.
- $ObjV^{next_i}_{MinSubscript}$: The fitness value of the worst individual in $next_i$th population.
- $MaxObjV$: The fitness values of the best individual in ROCKS.
- $MaxChromG$: The best individual topology that appears in ROCKS.

This algorithm works as follows. By using the initialization function (Algorithm 1), ROCKS assigns different topologies to each individual of each population based on the original scale-free WSNs topology (Line 1). In order to assign a pair of different crossover operator and mutation operator probabilities for different populations, a sliding variable is added to P_{cro} and P_{mut} (Lines 2 - 3). Before the optimization starts, the initial fitness values of each population are calculated (Lines 5 - 7). Then the ROCKS begins. The crossover operator (Algorithm 2) and mutation operator (Algorithm 3) are performed on each population in turn (Lines 9 - 12), and we get the fitness values $ObjVSel$ of current population (Line 13) by $ObjectFunction$. For the $genSafe$th generation of ith population, we put it in the $(genSafe-1)$th generation. And according to their fitness values, the best individuals are selected as the $genSafe$th population by the function $ReiCh$ (Line 14), which guarantees that the outstanding individuals of two generations can be retained. The best individual for each population of each generation is stored in $maxObjV$ (Line 15). When all the populations have performed the above operations in one generation, the migration operation begins.

All populations will be traversed (Line 17). We find out the index $MaxSubscript$ of the individual with the best fitness value in ith population by the function $MaxPo$ (Line 18), and the index $MinSubscript$ of the individual with the worst fitness value in $(i+1)$th population by the function $MinPo$ (Line 19 - 23). Then we use the $MaxSubscript$th individual topology of ith population to replace the $MinSubscript$th individual topology of $(i+1)$th population (Line 24 - 25). After migration operation, the best individual that appear in the current populations is searched by the function $EliteIndul$. $MaxObjV$ and $MaxChromG$ preserves its fitness values and topology (Line 27). This process continues until the number of generations $genSafe$ reaches $MaxGen$. Finally, the topology of the best individual that appear in the last generations $MaxChromG$ is assigned to G_R (Line 30).

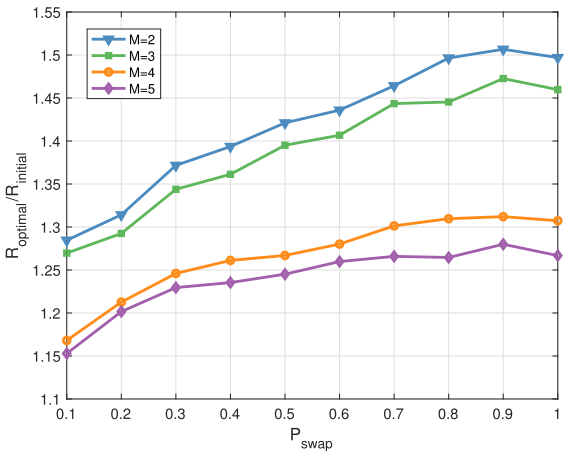


Fig. 7. The effect of parameter P_{swap} on the performance of ROCKS.

V. Simulation Results

We simulate ROCKS in MATLAB. The nodes are deployed randomly in a disk area with a diameter equal to $500m$. Considering that each node must have sufficient neighbors in the initial topology, the communication range is set to $200m$, and each node has 2 edges. We set the number of populations M_{pop} as 10, and the number of individuals in a population N_{ind} as 20. These two parameters are determined to be optimal for ROCKS by a large number of experiments.

A. The Threshold of Swapping P_{swap} in Different Edge Densities

The threshold of swapping edges P_{swap} is determined by experiment. The parameters of experiment are set as follows: $N = 100$ and $MaxGen = 200$, and the edge density M is $[2, 3, 4, 5]$. The value of P_{swap} slides from 0.1 to 1 and the interval is 0.1. Each round of experiments uses the same initial topology, and all results correspond to the average of $k(k > 10)$ independent runs. Finally, we get the trend of $R/R(init)$ with the change of P_{swap} value.

In Fig. 7, the x-axis represents the value of P_{swap}, and the y-axis represents the ratio of the optimized R with the initial R. With the increase of P_{swap} value, the effect of ROCKS optimization increases slowly, the optimal value is obtained at $P_{swap} = 0.9$ in different value of M. When the value of $P_{swap} = 1$, the optimization effect declines. Thus, we use $P_{swap} = 0.9$ in the next experiment.

B. Comparison Between Conventional GA and ROCKS

In this section, we make a comparison of the ROCKS and conventional GA. In order to observe a comprehensive effect under the same condition, the evolution of 10 independent populations in conventional GA is compared with ROCKS. The parameters of the experiment are set as follows: $N = 100$, $N_{ind} = 20$, $MaxGen = 200$. Besides, the probabilities of crossover operator and mutation operator in each single

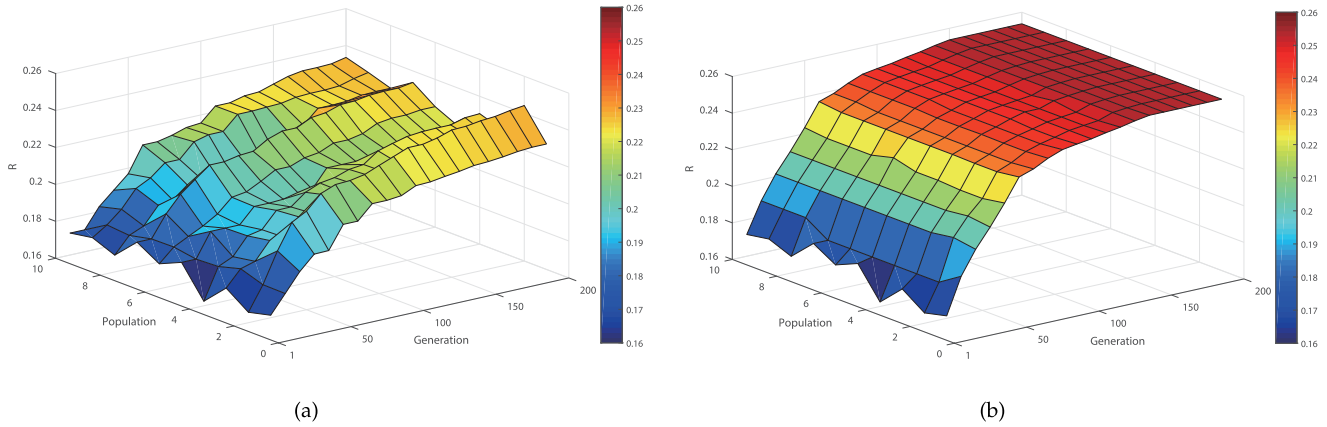


Fig. 8. The comparison between conventional GA and ROCKS. (a) Conventional GA. (b) ROCKS.

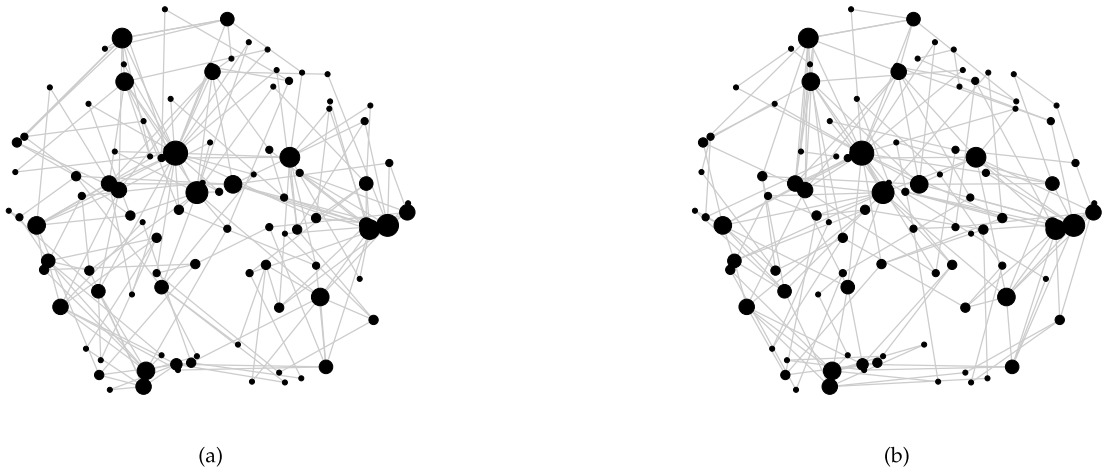


Fig. 9. Comparison between before and after ROCKS. (a) 100 nodes, before optimized, $R = 0.1686$. (b) 100 nodes, after optimized, $R = 0.2537$.

population are equal to those in multi-population. The initial topology of each population is identical in the two experiments. Fig. 8(a) shows the trend of the best individual fitness values of each generation in the evolution of 10 independent populations. And Fig. 8(b) shows the trend of the best individual fitness values of each generation in the evolution of ROCKS.

It can be seen from the experiments that the ROCKS achieves a great improvement in optimization compared with the conventional GA. As shown in Fig. 8(a), the evolution of each population is independent in the conventional GA. And the optimization effect of each population is affected by the values of crossover operator and mutation operator probabilities. The optimization results are very different, and the final optimal value also cannot reach a high level in the conventional GA. ROCKS (as shown in Fig. 8(b)) uses co-evolution and migration operations among populations. The excellent individuals are introduced to the other populations in every generation, which avoids falling into local optimum caused by an inappropriate pair of crossover and mutation operator probabilities. The above experiments show that through using the co-evolution of the ROCKS, premature convergence can be avoided effectively, and a better optimization result can be got.

Fig. 8(b) also illustrates that the metric R increases with the number of generations of ROCKS. At the beginning, the metric R of the initial topology is low, and the value of R increases obviously from 1st to 70th generation. After 70th generation, the optimization result increases slowly due to the value of R has increased to a high level.

C. Comparison Between Before and After ROCKS

In order to compare the network topology changes before and after the experiment, we designed the experiment in this section. The parameters of the experiment are set as follows: $N = 100$, $N_{ind} = 20$, $MaxGen = 200$.

As shown in Fig. 9, the total number of nodes in scale-free WSNs topology is 100. The initial topology is shown in Fig. 9 (a). The size of the node represents its degree, and the greater diameter of the node means that it has the greater degree. The metric R is 0.1686 before optimization, and the metric R increases to 0.2537 after optimization. We can see that the nodes which have similar degrees connect with each other in Fig. 9(b). Finally, our proposed algorithm makes the network topology close to the onion-like structure.

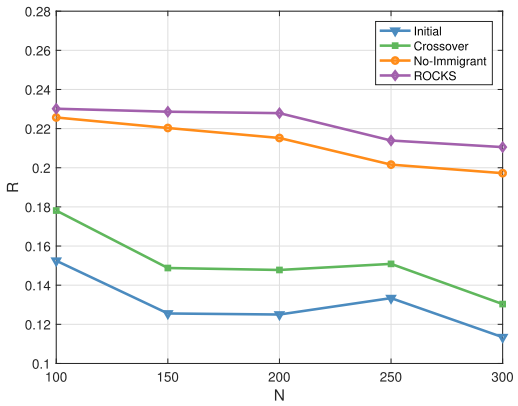


Fig. 10. The comparison between difference operations of ROCKS.

D. Comparison Among Difference Operations of ROCKS

we designed the experiment in this section to observe the effect of different operations in optimizing the robustness of network topology. The parameters of the experiment are set as follows: $N_{ind} = 20, MaxGen = 200$, and total number of nodes N is $[100, 150, 200, 250, 300]$.

Fig. 10 shows the comparison between difference operations of ROCKS. The blue line represents the initial network. The green line represents using crossover operator only. And the orange line represents using crossover operator and mutation operator. The purple line represents ROCKS, which include migration operator. As the number of nodes increases, the optimization effect shows a downward trend. Besides, all operators improve the optimization effect. Especially, when all operators are grouped together in ROCKS, the best optimization can be obtained.

E. Comparison on the Network Connectivity Before and After ROCKS

In order to observe the attack effect intuitively, the number of nodes in the maximally connected subgraph after removing the attacked node is used to measure the status of network connectivity. As shown in Fig. 11, the green line represents the initial topology, and the orange line represents the topology optimized by ROCKS. Besides, the blue line represents a fully connected network, in which all the nodes are connected to each other.

Random attack refers to that we select randomly nodes and remove all the edges with them. In order to observe the effect of random attack visually, we only attack a node each time until all the nodes have been attacked. The attack strategy is attacking a node randomly every second.

As shown in Fig. 11(a) and Fig. 11(b), After one random attack, the number of nodes in the maximal connected subgraph of fully connected network only reduces one. It can be seen that the two lines are basically consistent. After the optimization, the robustness of network against random attack is consistent with the initial topology, which means the characteristic of topology against random attack is not destroyed.

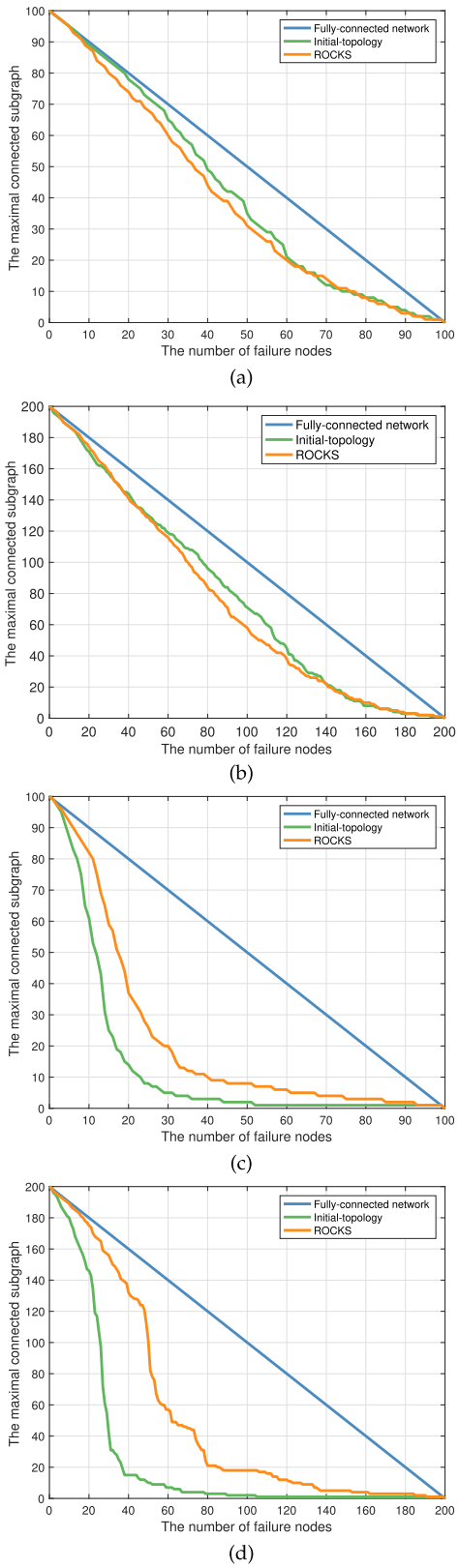


Fig. 11. Comparison on the network connectivity before and after ROCKS. (a) 100 nodes under random attacks. (b) 200 nodes under random attacks. (c) 100 nodes under malicious attacks. (d) 200 nodes under malicious attacks.

Malicious attack of network refers to attack the important nodes, which leads to the collapse of entire network within short time. Unlike the random attack, malicious attack is a

purposeful attack. We calculate the degrees of each node in the experiment. The bigger the degree of a node, which means that it connects with more nodes, the more important it is in the network. In order to simulate malicious attack, the node with the biggest degree is the first target to be attacked. The attack strategy is attacking a node purposefully every second. The initial topology and the optimized topology are compared under the same conditions against malicious attack, and the number of nodes in maximal connected subgraph after each malicious attack are recorded.

As shown in Fig. 11(c) and Fig. 11(d), the orange line is closer to the blue line, which means that the topology after optimized by ROCKS is closer to the fully connected network ubder against malicious attack. For malicious attacks, the optimized network is more robust than initial network. Thus, ROCKS can significantly enhance the robustness of network against malicious attack.

F. Comparison Between ROCKS and Other Algorithms in Different Edge Densities

In this subsection, we compare the ROCKS with the Hill Climbing algorithm [29] and the Simulated Annealing algorithm [27] for robust optimization under different edge densities. The parameters of the experiment are set as follows: $N = 100$, $M_{pop} = 10$, $N_{ind} = 20$, $MaxGen = 200$, and the edge density M is $[1, 2, 3, 4, 5]$. The edge density M represents the number of edges connected to each node in the original topology. The same comparison tests are in the same condition, which means that the initial topology is the same. All results correspond to the average of k $(k > 10)$ independent runs.

In the simulation of Hill climbing, each edge of topology is traversed. For every selected edge, the rest edges are traversed until find an edge in its communication range. The metric R of the new combinations in the two reconnected edges will be calculated. Select the combination in which metric R is increased to reconnect the two edges. If metric R do not increase, the two edges will not be swapped. The above operation will continue until all edges are traversed.In the simulation of Simulated Annealing, each edge of topology is traversed. For every selected edge, an edge in its communication range will be found in the reset edges. For the two selected edges, the metric R of the candidate edge-swap operation will be calculated. If metric R increases, we accept candidate edge-swap operation. If metric R is decreased, candidate edge-swap operation will be accepted in a determined probability T, which is called simulation temperature and is set as 0.001. If the decrease in metric R is accepted, the value of probability T will be halved.

It can be seen from Fig. 12 that with the increase of the edge density of the scale-free WSNs, all the three algorithms improve the metric R over the initial network topology significantly, which means the ability of the network topology against malicious attacks increases gradually. With the increase of network edges density, both the three algorithms present a similar trends. The optimization compared with the initial topology is not obvious when the edge density is too large or too small. A appropriate edge density can make the optimization effects of the three algorithms more easy to observe. Besides, for different edges densities of scale-free WSNs topology, ROCKS always has a better optimization results compared with the traditional optimization algorithms.

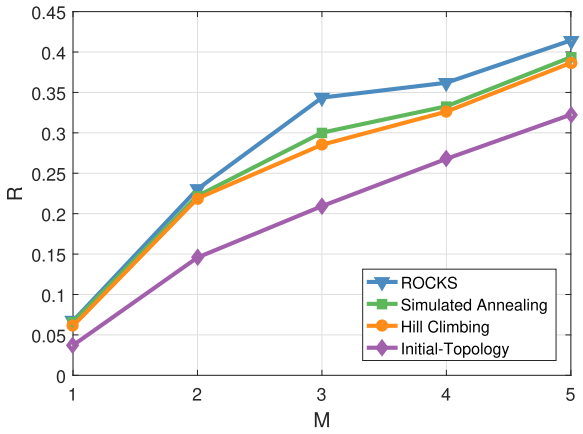


Fig. 12. Comparison between ROCKS and other algorithms in different edge densities.

G. Comparison Between ROCKS and Other Algorithms in Different Network Sizes

Based on the original scale-free WSNs topology, we compare our algorithm with two existing algorithms, namely Hill Climbing algorithm and Simulated Annealing algorithm.

The working principles of Hill Climbing and Simulated Annealing are briefed as follows: Hill Climbing belongs to local search algorithms. The Hill Climbing algorithm only compares with the previous optimization state and the next optimization state. When it is better than the previous state and the next state, it considers the current state as the optimal solution. Although the efficiency of Hill Climbing is relatively high, the quality of the optimal solution is relatively poor. Simulated Annealing introduces random factors to its search process. Simulated Annealing accepts a solution that is worse than the current solution with a certain probability, so it is possible to jump out of this local optimal solution and reach a global optimal solution.

Both of these algorithms keep the initial degree of every node unchanged in experiment. Fig. 13 shows that the optimization results of Hill Climbing algorithm, Simulated Annealing algorithm and our proposed algorithm in different sizes of scale-free WSNs topology. The size of scale-free topologies is set as $100, 150, 200, 250, 300$ nodes, respectively. The results are the average of k $(k > 10)$ independent experiments and each scale-free WSNs topology remains connected after optimization.

As can be seen in Fig. 13, all the three algorithms improve the robustness of the initial topology significantly. The performance of the Simulated Annealing algorithm is better than the Hill Climbing algorithm. All the three algorithms present a downward trend with the increase of network sizes. ROCKS always has the best performance results in robustness optimization than the other two algorithms in WSNs.

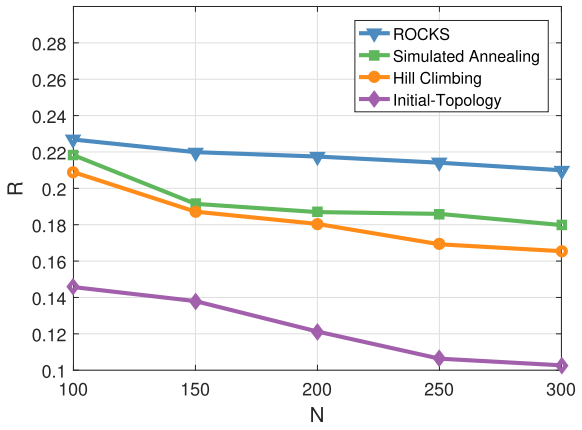


Fig. 13. Comparison between ROCKS and other algorithms in different network sizes.

VI. Conclusion

Due to the characteristics of WSNs, such as limited communication range and limited node degree, the traditional BA model in wired network is no longer suitable. We first gave an adapted method from the BA model to construct scale-free topologies for WSNs. Then, we proposed a scheme called ROCKS to optimize the robustness of the constructed scale-free topologies. In this ROCKS, we designed two novel operators, namely crossover operator and mutation operator. Under the evolution of these two operators, the initial degree of each node is unchanged, thus the scale-free property is preserved. Finally, we simulate our algorithm and two existing algorithms on their performances in improving the robustness of scale-free WSNs topologies under different edge densities and network sizes. The experiment results show that our algorithm can significantly improve the robustness of scale-free WSNs against malicious attacks. With the network size increasing, the values of R in two existing algorithms drop quickly, but ROCKS still maintains the values of R at a high level.

ROCKS is designed for enhancing the robustness of WSNs in a centralized system, which needs the information of the entire scale-free network topology.We will further focus on WSNs in a distributed system, and explore the application of multi-population co-evolution algorithm in it.

References

[1] S. Ji, R. Beyah, and Z. Cai, "Snapshot and continuous data collection in probabilistic wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 626–637, Mar. 2014.

[2] F. M. Al-Turjman, H. S. Hassanein, and M. Ibnkahla, "Towards prolonged lifetime for deployed WSNs in outdoor environment monitoring," *Ad Hoc Netw.*, vol. 24, pp. 172–185, Jan. 2015.

[3] Y. Yao, Q. Cao, and A. V. Vasilakos, "EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 810–823, Jun. 2015.

[4] A. A. M. Rahat, R. M. Everson, and J. E. Fieldsend, "Evolutionary multi-path routing for network lifetime and robustness in wireless sensor networks," *Ad Hoc Netw.*, vol. 52, pp. 130–145, Dec. 2016.

[5] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, Mar. 2001.

[6] N. Robertson, P. Seymour, and R. Thomas, "Quickly excluding a planar graph," *J. Combinat. Theory, B*, vol. 62, no. 2, pp. 323–348, Nov. 1994.

[7] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Networks formed from interdependent networks," *Nature Phys.*, vol. 8, no. 1, pp. 40–48, 2012.

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

[9] K. Wei, S. Guo, D. Zeng, K. Xu, and K. Li, "Exploiting small world properties for message forwarding in delay tolerant networks," *IEEE Trans. Comput.*, vol. 64, no. 10, pp. 2809–2818, Oct. 2015.

[10] T. Qiu *et al.*, "A greedy model with small world for improving the robustness of heterogeneous Internet of Things," *Comput. Netw.*, vol. 101, pp. 127–143, Jun. 2016.

[11] D. Luo, T. Qiu, N. Deonauth, and A. Zhao, "A small world model for improving robustness of heterogeneous networks," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2015, pp. 849–852.

[12] T. Qiu, A. Zhao, F. Xia, W. Si, and D. O. Wu, "ROSE: Robustness strategy for scale-free wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2944–2959, Oct. 2017.

[13] E. Bulut and B. K. Szymanski, "Constructing limited scale-free topologies over peer-to-peer networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 919–928, Apr. 2014.

[14] R.-H. Li, J. X. Yu, X. Huang, H. Cheng, and Z. Shang, "Measuring robustness of complex networks under MVC attack," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2012, pp. 1512–1516.

[15] H. Ren, X.-N. Huang, and J.-X. Hao, "Finding robust adaptation gene regulatory networks using multi-objective genetic algorithm," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 3, pp. 571–577, May/Jun. 2016.

[16] H. M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to prevent premature convergence in GA," *Appl. Soft Comput.*, vol. 24, pp. 1047–1077, Nov. 2014.

[17] W. Xiao, L. Lin, and G. Chen, "Vertex-degree sequences in complex networks: New characteristics and applications," *Phys. A, Stat. Mech. Appl.*, vol. 437, pp. 437–441, Nov. 2015.

[18] X. Wang, S. Han, Y. Wu, and X. Wang, "Coverage and energy consumption control in mobile heterogeneous wireless sensor networks," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 975–988, Apr. 2013.

[19] Z. Zheng, A. Liu, L. X. Cai, Z. Chen, and X. Shen, "Energy and memory efficient clone detection in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1130–1143, May 2016.

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

[21] R. N. Shukla, A. S. Chandel, S. K. Gupta, J. Jain, and A. Bhansali, "GAE^{3}BR: Genetic algorithm based energy efficient and energy balanced routing algorithm for Wireless Sensor Networks," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Aug. 2015, pp. 942–947.

[22] M. Elhoseny *et al.*, "Balancing energy consumption in heterogeneous wireless sensor networks using genetic algorithm," *IEEE Commun. Lett.*, vol. 19, no. 12, pp. 2194–2197, Dec. 2015.

[23] A. Peiravi, H. R. Mashhadi, and S. H. Javadi, "An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm," *Int. J. Commun. Syst.*, vol. 26, no. 1, pp. 114–126, Jan. 2013.

[24] M. Zhou and J. Liu, "A memetic algorithm for enhancing the robustness of scale-free networks against malicious attacks," *Phys. A, Stat. Mech. Appl.*, vol. 410, pp. 131–143, Sep. 2014.

[25] C. M. Schneider, A. A. Moreira, J. S. Andrade, Jr., S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 10, pp. 3838–3841, 2011.

[26] H. Hu, G. Li, and J. Feng, "Fast similar subgraph search with maximum common connected subgraph constraints," in *Proc. IEEE Int. Congr. Big Data*, Jun./Jul. 2013, pp. 181–188.

[27] P. Buesser, F. Daolio, and M. Tomassini, "Optimizing the robustness of scale-free networks with simulated annealing," in *Proc. Int. Conf. Adapt. Natural Comput. Algorithms*. Berlin, Germany: Springer, 2011, pp. 167–176.

[28] V. H. P. Louzada, F. Daolio, H. J. Herrmann, and M. Tomassini, "Smart rewiring for network robustness," *J. Complex Netw.*, vol. 1, no. 2, pp. 150–159, Dec. 2013.

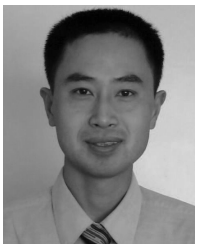
[29] H. J. Herrmann, C. M. Schneider, A. A. Moreira, J. S. Andrade, and S. Havlin, "Onion-like network topology enhances robustness against malicious attacks," *J. Stat. Mech., Theory Exp.*, vol. 2011, no. 1, pp. 1–9, Jan. 2011.

[30] T. Qiu *et al.*, "A data-driven robustness algorithm for the Internet of Things in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 18–23, Dec. 2017.

[31] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 5, pp. 1–15, May 2002.

Tie Qiu (M'12–SM'16) received the Ph.D. degree in computer science from the Dalian University of Technology in 2012. He was an Assistant Professor and an Associate Professor with the School of Software, Dalian University of Technology, in 2008 and 2013, respectively. He was a Visiting Professor in electrical and computer engineering with Iowa State University, USA, from 2014 to 2015. He is currently a Full Professor with the School of Computer Science and Technology, Tianjin University, China. He has authored or coauthored nine books, more than 100 scientific papers in international journals and conference proceedings, such as the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and the *IEEE Communications Magazine*. There are ten papers listed as ESI highly cited papers. He has contributed to the development of three copyrighted software systems and invented 12 patents. He is a Senior Member of the China Computer Federation (CCF) and a Senior Member of the ACM. He serves as a General Chair, a Program Chair, a Workshop Chair, a Publicity Chair, a Publication Chair, or a TPC Member of a number of international conferences. He serves as an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, an Area Editor of *Ad Hoc Networks*, an Associate Editor of the IEEE ACCESS, *Computers and Electrical Engineering*, and *Human-Centric Computing and Information Sciences*, and a Guest Editor of *Future Generation Computer Systems*.

Jie Liu received the B.E. degree from the Dalian University of Technology (DUT), China, in 2015. He is currently pursuing the master's degree with the School of Software, DUT. He participated in the Embedded Internet of Things Competition 2014 and received the National third prize. He obtained an invention patent in 2016. He received several scholarships in academic excellence and technology innovation. He is a member of the Smart Cyber-Physical Systems Laboratory (SmartCPS Lab). His research interests cover robustness optimization of wireless sensor networks.



Weisheng Si (M'08) received the B.S. degree from the University of Sydney, Australia, the M.S. degree from the University of Virginia, USA, and the Ph.D. degree from Peking University, China, all in computer science. He is currently a Senior Lecturer with the School of Computing, Engineering and Mathematics, Western Sydney University. He was a Post-Doctoral Researcher with the National ICT Australia. His research interests mainly include cybersecurity, software defined networking, complex networks, and Internet of Things.

Dapeng Oliver Wu (S'98–M'04–SM'06–F'13) received the B.E. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.E. degree in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security. He received the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (CSVT) Best Paper Award for Year 2001, the International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006, the NSF CAREER Award in 2007, the ONR Young Investigator Program (YIP) Award in 2008, the University of Florida Research Foundation Professorship Award in 2009, the AFOSR Young Investigator Program (YIP) Award in 2009, the Best Paper Awards in IEEE GLOBECOM 2011, and the University of Florida Term Professorship Award in 2017.

Prof. Wu was an Elected Member of the Multimedia Signal Processing Technical Committee, the IEEE Signal Processing Society, from 2009 to 2012. He served as a member of the executive committee and/or technical program committee (TPC) of more than 100 conferences. He has served as a Chair for the Award Committee and a Chair of the Mobile and Wireless Multimedia Interest Group (MobIG), Technical Committee on Multimedia Communications, and the IEEE Communications Society. He has served as a TPC Chair for IEEE INFOCOM 2012 and the IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium. He currently serves as an Editor-in-Chief for the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. He was the founding Editor-in-Chief of the *Journal of Advances in Multimedia* from 2006 to 2008, and an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, the *IEEE Signal Processing Magazine*, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is also a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Special Issue on Cross-layer Optimized Wireless Multimedia Communications and Special Issue on Airborne Communication Networks. He was elected as a Distinguished Lecturer by the IEEE Vehicular Technology Society in 2016.