

## Research Article

# Ripple-Spreading Network Model Optimization by Genetic Algorithm

Xiao-Bing Hu,<sup>1,2</sup> Ming Wang,<sup>1</sup> and Mark S. Leeson<sup>2</sup>

<sup>1</sup> State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, 100875 Beijing, China

<sup>2</sup> School of Engineering, University of Warwick, Coventry CV4 7AL, UK

Correspondence should be addressed to Xiao-Bing Hu; [dr\\_xiaobinghu@hotmail.co.uk](mailto:dr_xiaobinghu@hotmail.co.uk)

Received 3 May 2013; Accepted 19 July 2013

Academic Editor: Vishal Bhatnagar

Copyright © 2013 Xiao-Bing Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Small-world and scale-free properties are widely acknowledged in many real-world complex network systems, and many network models have been developed to capture these network properties. The ripple-spreading network model (RSNM) is a newly reported complex network model, which is inspired by the natural ripple-spreading phenomenon on clam water surface. The RSNM exhibits good potential for describing both spatial and temporal features in the development of many real-world networks where the influence of a few local events spreads out through nodes and then largely determines the final network topology. However, the relationships between ripple-spreading related parameters (RSRPs) of RSNM and small-world and scale-free topologies are not as obvious or straightforward as in many other network models. This paper attempts to apply genetic algorithm (GA) to tune the values of RSRPs, so that the RSNM may generate these two most important network topologies. The study demonstrates that, once RSRPs are properly tuned by GA, the RSNM is capable of generating both network topologies and therefore has a great flexibility to study many real-world complex network systems.

## 1. Introduction

As is well known, many successful artificial intelligence technologies are actually inspired by certain natural systems or phenomena [1–3]. For instance, genetic algorithms (GAs) are inspired by natural selection and evolutionary processes, artificial neural networks (ANNs) by the animal brain, particle swarm optimization (PSO) by the learning behavior within a population, ant colony optimization (ACO) by the foraging behavior of ants, and optimal structure design by the hexagonal cells of the honeycomb.

Following the common practice of learning from nature in the artificial intelligence domain, we have recently proposed a ripple-spreading network model (RSNM) [4], which is inspired by the natural ripple-spreading phenomenon. A ripple-spreading process defined by some ripple-spreading related parameters (RSRPs) is the central piece of the model. Basically, assuming that nodes of networks are distributed in a (real or artificial) space and that the distribution of nodes are fixed, some initial ripples are randomly generated in the space. As an initial ripple spreads out in the space, its point

energy decays gradually, and it reaches every node one by one sooner or later. By comparing the point energy of an incoming ripple with some preset thresholds for a node, it can be determined whether this node will be activated by the incoming ripple to generate a new ripple, and whether this node will be connected to the node where the incoming ripple originates. When a node is activated to generate a new ripple, the initial energy of the new ripple will be a function of the point energy of the incoming ripple. New ripples are also able not only to activate other nodes to generate more ripples but also to establish new connections between nodes, as long as their point energy is above the relevant thresholds. As this ripple-spreading process goes on for a while, a network topology will appear. Actually, the output topology will be largely determined by the values of RSRPs, such as the locations of the epicenters for initial ripples, the thresholds to tell whether a node will be activated or connected, the energy amplifying factor, and the coefficients to define the point energy decaying rate. Once the values for these RSRPs are given and fixed, then the output topology will be uniquely determined.

However, when compared with many existing network models such as those reported in [5, 6], there is a lack of obvious or straightforward relationships between the RSRPs and the network properties of RSNM topology. In particular, [4] listed it as one of the future works to study when and how the RSNM may generate two of the most important network topologies, small-world and scale-free topologies. Therefore, based on the theoretical and conceptual results in [4], this paper focuses in particular on providing a practicable method to tune the RSRPs of RSNM, in order to generate preferable network topologies. To this end, the model in [4] will be modified into a more general form to ease the analysis on the properties of RSNM, and then an effective GA will be developed, in order to tune the RSNM to generate small-world and scale-free network topologies.

The remainder of this paper is organized as follows. Section 2 gives some background knowledge of related work. Section 3 describes a mathematical model of the modified RSNM. The proposed GA will be explained in Section 4 for optimizing the values of RSRPs. Section 5 reports some simulation results. The paper ends with some conclusions and discussions on future work in Section 6.

## 2. Related Work on Complex Networks

Complex networks, that is, networks whose structure is irregular, complex, and dynamically evolving in time, describe a wide range of systems in nature and society and are all around us in our daily life [6, 7]. In the past few decades, many efforts have been made to model and analyze various complex networks. Most network models can be classified as stochastic models because they have a typical feature in common: a stochastic model abstracts one or a few network properties or parameters to capture in quantitative terms the underlying organizing principles of complex networks, and these network properties or parameters can estimate how the network appears but cannot guarantee an exact or unique topology. In other words, different network topologies may have exactly the same values for the specified properties or parameters. For instance, in classic random graph theory [5], the connection probability is the core parameter of the network model. In the generalized random graph model [8], a degree distribution following a power law is used as the input in order to be able to describe the scale-free character of real networks that the classic random graph model cannot capture. In the theory of evolving networks [9], the parameter known as the preferential attachment is often used to model scale-free real networks. In a recently reported spatially embedded random network model [10], the connection probability is formulated as a function of distance between nodes. In either of the above network models, even if the input is fixed, for example, the connection probability, power law for degree distribution or preferential attachment is fixed, the output of the model is enormous, and what the output topology exactly looks like is largely by chance. Therefore, they are all stochastic models. With the exception of those network properties which explicitly depend on the input parameters, the output topology of the models is

largely unpredictable or uncertain in terms of other network properties. As a result, it is difficult to apply these models in network design, such as the topology optimization problem.

In contrast, the RSNM proposed in [4] is a deterministic complex network model. Unlike the stochastic models in [5, 8–10], once the values of input are fixed, the output network topology will also be fixed and unique in the new model. Therefore, we can adjust these RSRPs in order to improve the network topology in terms of concerned network properties. Furthermore, there is great freedom and flexibility to modify and extend the RSNM. For instance, the proposed RSNM can easily be extended to a semideterministic version and a stochastic version. Basically, the RSNM can introduce randomness at two levels by RSRPs: (i) the random setup of nodes (e.g., the random spatial distribution of nodes and the random distribution of node thresholds and amplifying factors, if they are not constants); (ii) the random setup of initial stimulating ripples (e.g., locations, initial energy, and starting time). The first level of randomness is related to the internal factors of the system, and the second level of randomness is associated with external impacts to the system. The two levels of randomness in the RSNM make it very easy and natural for the RSNM to capture/reflect both internal and external factors in many real-world complex networks. For instance, the impact of earthquake through a risk chain and the breakout of plagues in a community can be described as evolving networks triggered by a few initial simulating ripples; the threshold of nodes is related to the vulnerability of infrastructure and the immunity of individuals and the amplifying factor of nodes to connectivity of infrastructure and the social activeness of individuals.

Any complex network model claimed to be practicable must illustrate the capability of generating network topologies that present in real-world systems, among which small-world and scale-free topologies are two of the most important network topologies. It is widely recognized that many real-world complex network systems display some organizing principles, which often lead to either small-world topology or scale-free topology [7, 11, 12]. The small-world concept in simple terms describes the fact that despite their often large size, in most networks, there is a relatively short path between any two nodes. The distance between two nodes is defined as the number of edges along the shortest path connecting them. The most popular manifestation of small worlds is the “six degrees of separation” concept, which concluded that there was a path of acquaintance with a typical length of about six between most pairs of people in the United States [13]. The small-world property appears to characterize most complex networks, for instance, the actors in Hollywood are on average within three costars from each other or the chemicals in a cell are typically separated by three reactions. Another most important development in our understanding of complex networks was the discovery that for a large number of networks, including the World Wide Web [14], the Internet [15], or metabolic networks [16], the degree distribution has a power-law tail. Such networks are called scale free [17]. While some networks display an exponential tail, often the functional form of degree distribution still deviates significantly from the Poisson distribution expected

for a random graph. The discoveries of small-world and scale-free features have fundamentally reshaped the study of complex network systems. Therefore, to be able to better apply the RSNM in [4] to study real-world complex networks, it is crucial to understand how and when the RSNM will generate small-world and scale-free network topologies, which is the focus of this paper.

### 3. Mathematical Description of Ripple-Spreading Network Model (RSNM)

**3.1. The Basic Idea of RSNM.** The RSNM was first reported in [4], where the basic idea was inspired by the natural ripple-spreading phenomenon on calm water surfaces. Suppose that a collection of stakes is randomly distributed in a quiet pond. Then, suddenly a stone is thrown into the pond, and an initial ripple is generated and spreads out from the point where the stone hits the quiet water. When the ripple reaches a nearby stake, a new ripple is generated around the stake due to the reflection effect. For the sake of consistency, we call such a new ripple a *responding* ripple (or outgoing ripple), and the ripple which triggers the responding ripple as a *stimulating* ripple (or incoming ripple). Obviously, the initial energy of a responding ripple is determined by both the point energy of the stimulating ripple and the physical features of the stake (e.g., whether the stake is rigid or soft in material). A responding ripple may trigger new responding ripples around other stakes. As the initial stimulating ripple spreads out, more and more responding ripples are stimulated around other stakes. However, since the point energy on the stimulating ripples decays as they spread out, those responding ripples triggered at a late phase could hardly be noticed. Therefore, we may say there is a threshold, and only when the point energy of a stimulating ripple is above this threshold will a responding ripple be generated. Furthermore, the reaction of stake to a stimulating ripple may be more than just generating a responding ripple. For instance, if the point energy of a stimulating ripple is above another threshold, the stake will shake noticeably to disturb a bird resting on the top of the stake. Once the bird on a stake is disturbed by the outgoing ripple of another stake, we set up an edge between the associated pair of nodes in a network. As the time elapses, the point energy of all ripples will decay to below the thresholds sooner or later, and after a long enough period of time, all ripples will disappear and no birds will be disturbed any more. Then, based on the established edges according to which bird has been disturbed by the outgoing ripple of which stake, we will get a network topology. Clearly there are some factors affecting the final network topology. For instance, how many stones hit the pond to generate initial stimulating ripples, where do they hit the pond, what is the mass of each stone (will determine the initial energy of the associated stimulating ripple), what are the preset values for the thresholds, and what is the amplifying factor of each stake to reflect a stimulating ripple? By mathematically formulating these factors and the relationships between them, we can get an effective model for complex networks. For more details of the basic idea of RSNM, readers may refer to [4].

**3.2. Mathematical Model of Modified RSNM.** The RSNMs proposed in [4] have an important concept: epicenter of initial stimulating ripple (EISR), which is associated with the point where the stone hit the water surface. Unlike normal nodes in network, an EISR is not connectable, which means no edge can be established between two EISRs or an EISR and a normal node. EISRs are useful to match and describe the initial cause of the development of a real-world network; for instance, in the study on the impact of earthquake on a transport network, the epicenter of earthquake is usually not located at any node of the transport network. However, the difference between EISRs and normal nodes makes it very difficult to conduct theoretical deduction of network properties. This partially explains why paper [4] failed to analyze the small-world and scale-free properties of the RSNM. Without necessary theoretical analyses, the RSNM can hardly develop into a mature and practicable method. From a simulation point of view, a network model with EISRs can be mathematically equivalent to a model without EISRs, and the latter will be much easier to analyze and study. Therefore, in this paper, we will modify the models in [4] by removing EISRs and adding a new concept of self-generating ripple (SGR). Basically, in the modified RSNM all initial stimulating ripples will be self-generated by connectable nodes in the network, in other words, all initial stimulating ripples will directly come from connectable nodes. To this end, two new ripple-spreading related parameters (RSRPs) need to be introduced to every node: initial energy of SGR  $E_{\text{SGR}}(i)$  and start time of SGR  $T_{\text{SGR}}(i)$ ,  $i = 1, \dots, N_N$ , where  $N_N$  is the total number of nodes in the network. By presetting  $E_{\text{SGR}}(i)$  and  $T_{\text{SGR}}(i)$  according to the impact of EISRs, one can easily see that the modified RSNM is equivalent to the models reported in [4]. Actually, the modified RSNM is a generalized model, as the models in [4] are just a subset of the new model. Obviously, only those nodes with  $E_{\text{SGR}}(i) > 0$  matter in terms of EISRs, and we assume that there are  $N_{\text{SGR}} \leq N_N$  such nodes.

Before we give the mathematical description of the modified RSNM, we need to define some other important ripple-spreading related parameters (RSRPs) like in [4]. For the sake of simplicity but without loss of generality, unless explicitly specified, we suppose all nodes are distributed in a limited 2-dimensional space in this paper. It is also assumed that the total number of nodes,  $N_N$ , and their locations,  $(x_N(i), y_N(i))$ , are already given and fixed. To get different topologies from this fixed set of nodes, we have another three RSRPs to each node:  $\alpha(i)$ ,  $\beta_R(i)$ , and  $\beta_L(i)$ , which are the amplifying factor, the threshold to generate a responding ripple, and the threshold to establish a link, for node  $i$ ,  $i = 1, \dots, N_N$ , respectively.

With the above RSRPs, the modeling process of the modified RSNM can be mathematically described as follows.

*Step 1.* Initialize the current time instant, that is,  $t = 0$ . Initialize  $E_{\text{SGR}}(i)$  and  $T_{\text{SGR}}(i)$  for every node randomly or according to certain requirements. Since each node has no initial energy, that is,  $E_N(i) = 0$ , therefore, its current point energy is

$$e_N(i, t) = E_N(i) = 0, \quad i = 1, \dots, N_N. \quad (1)$$

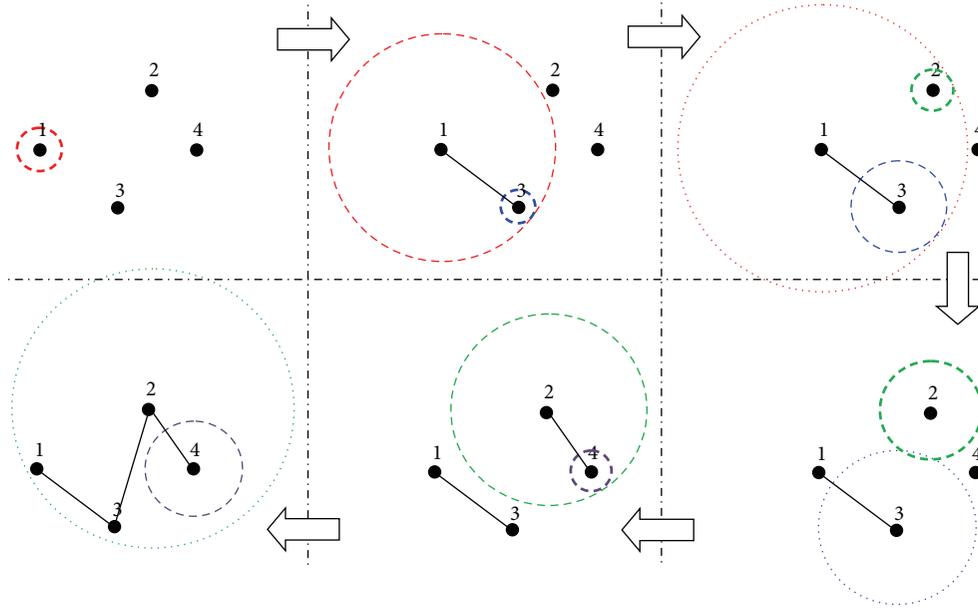


FIGURE 1: The development of a simple network topology based on the RSNM, where the epicenter of initial stimulating ripple (EISR) is a connectable node in the network, that is, node 1.

Assume that each node has a ripple with a current radius of 0, that is,  $r_N(i, t) = 0$ .

*Step 2.* If the termination criteria are not satisfied, do as follows.

*Substep 1.* Let  $t = t + 1$ .

*Substep 2.* Check  $t$  against  $T_{SGR}(i)$ . If  $t \geq T_{SGR}(i)$ ,  $E_{SGR}(i) > 0$ , and  $r_N(i, t) = 0$ , then the self-generated ripple of node  $i$  will start to spread, so set up the initial energy of the ripple starting from node  $i$  as

$$E_N(i) = E_{SGR}(i). \quad (2)$$

*Substep 3.* If  $E_N(i) > 0$ ,  $i = 1, \dots, N_N$ , then update the current radius and point energy of the ripple starting from node  $i$

$$\begin{aligned} r_N(i, t) &= r_N(i, t-1) + s, \\ e_N(i, t) &= f_{\text{Decay}}(E_N(i), r_N(i, t), t), \end{aligned} \quad (3)$$

where  $s$  is the spreading speed of ripples, that is, the change in the radius of a ripple during one time instant, and  $f_{\text{Decay}}$  is a function defining how the point energy decays as the ripple spreads out. A typical decaying function may be

$$f_{\text{Decay}}(E_N(i), r_N(i, t), t) = \eta \frac{E_N(i)}{2\pi r_N(i, t)}, \quad (4)$$

where  $\eta$  is a coefficient and  $\pi$  is the mathematical constant. Clearly  $\eta$  has an important influence on the decaying speed of ripples and will therefore affect the final network topology. Basically, (3) show that the ripples have the same spreading speed and the same decaying function, just like in the natural

world. However, this may be amendable in order to get a more complicated artificial model.

*Substep 4.* Check which new nodes are reached by the ripples. Suppose that  $L_N(i, j)$  is the distance between node  $i$  and node  $j$ . If  $E_N(j) = 0$  and  $L_N(i, j) \leq r_N(i, t)$ , then node  $j$  is reached by the ripple generated by node  $i$ . If  $e_N(i, t) \geq \beta_R(j)$ , then node  $j$  is activated by node  $i$  and generates a responding ripple with

$$E_N(j) = \begin{cases} \alpha(j) e_N(i, t), & E_{SGR}(j) \leq \alpha(j) e_N(i, t), \\ E_{SGR}(j), & E_{SGR}(j) > \alpha(j) e_N(i, t). \end{cases} \quad (5)$$

If  $e_N(i, t) \geq \beta_L(j)$ , then a connection between node  $i$  and node  $j$  is established, that is,

$$A(i, j) = A(j, i) = 1, \quad (6)$$

where  $A$  is the adjacency matrix which records the network topology.

Different termination criteria may be used in Step 2. For instance, the current time instance is beyond a specific time window, no node has current point energy above any threshold, or the upper bound for the number of total links is reached.

Figure 1 gives a simple illustration of the development of network topology based on the modified RSNM, where the initial stimulating ripple directly comes from a connectable node in the network. From Figure 1, one may notice that since the threshold for triggering a responding ripple may differ from the threshold for establishing an edge, a responding ripple is not always accompanied by the establishment of an edge. For instance, the outgoing ripple of node 1 triggers

a responding ripple at node 3 accompanied by the establishment of an edge to node 2, it only triggers a responding ripple at node 2, and it does nothing to node 4.

**3.3. Some Important Conditions.** We have previously presented some conditions for analyzing general network properties of the EISR-based RSNM [4], and here, we need to work out new conditions for the modified RSNM based on the concept of SGR. The proofs of new conditions are similar to those in [4], to which readers can refer to for details. Such new conditions will be very useful as heuristic rules in the application of GA to optimize the output network topology of RSNM.

For a network with  $N_N$  nodes, the number of total potential edges between nodes is

$$N_{PE} = \frac{N_N \times (N_N - 1)}{2}. \quad (7)$$

For each pair of nodes  $(i, j)$ , let  $L_N(i, j)$  denote the length of the potential edge between node  $i$  and node  $j$ , that is, the direct distance between node  $i$  and node  $j$ , no matter whether or not there is an actual connection between the pair of nodes. Since all nodes are distributed in a limited space, we can assume that  $L_U$  and  $L_L$  are the maximal and minimal lengths in all potential edges, respectively,  $E_{SGR,U}$  and  $E_{SGR,L}$  the maximal and minimal values of all initial SGR energy  $E_{SGR}(i)$ ,  $\beta_{R,U}$  and  $\beta_{R,L}$  the maximal and minimal values of all activating threshold  $\beta_R(i)$ ,  $\beta_{L,U}$  and  $\beta_{L,L}$  the maximal and minimal values of all connecting threshold  $\beta_L(i)$ , and  $\alpha_U$  and  $\alpha_L$  maximal and minimal values of all amplifying factor  $\alpha(i)$ ,  $i = 1, \dots, N_N$ .

**Condition 1** (completely connected graph). If the inequalities

$$\eta \frac{E_{SGR,U}}{2\pi L_{\text{MaxMin}}} \geq \beta_{R,U}, \quad (8)$$

$$\eta \frac{\alpha_L \beta_{R,L}}{2\pi L_U} \geq \max(\beta_{R,U}, \beta_{L,U}) \quad (9)$$

hold, where

$$L_{\text{MaxMin}} = \max_{i=1, \dots, N_N} \min_{j=1, \dots, N_N} L_N(i, j), \quad (10)$$

then the RSNM will generate a completely connected graph.

Equation (8) guarantees that the SGRs will trigger at least one responding ripple, and (9) guarantees that the responding ripple will cause a completely connected graph.

**Condition 2** (all nodes connected). If (8) and the inequality

$$\eta \frac{\alpha_L \beta_{R,L}}{2\pi L_{\text{TSP}}} \geq \max(\beta_{R,U}, \beta_{L,U}) \quad (11)$$

hold, where  $L_{\text{TSP}}$  denotes the maximum edge length in a shortest open travelling salesman problem (TSP) route, then the RSNM will generate a graph where all nodes are connected.

**Condition 3** (all nodes activated). If (8) and the inequality

$$\eta \frac{\alpha_L \beta_{R,L}}{2\pi L_{\text{TSP}}} \geq \beta_{R,U} \quad (12)$$

hold, then every node will generate an outgoing ripple under the RSNM.

The exact node degree can be counted as follows: for node  $i$ ,  $i = 1, \dots, N_N$ , its degree is

$$D_N(i) = \sum_{n=1}^{N_N} \sum_{m=n+1}^{N_N} A(n, m), \quad (13)$$

where  $A$  is the adjacency matrix given in (9) and its entries are determined by the ripple-spreading process described in Section 3.2. After the ripple-spreading process terminates, one has

$$A(n, m) = \begin{cases} 1, & \frac{\eta E_N(n)}{2\pi L_N(n, m)} \geq \beta_L(m) \\ & \text{or } \frac{\eta E_N(m)}{2\pi L_N(n, m)} \geq \beta_L(n), \\ 0, & \frac{\eta E_N(n)}{2\pi L_N(n, m)} < \beta_L(m), \\ & \frac{\eta E_N(m)}{2\pi L_N(n, m)} < \beta_L(n). \end{cases} \quad (14)$$

Please note that (13) and (14) are used to calculate the exact degree of each node after a network topology is generated by the RSNM. To roughly estimate node degree distribution from the RSRPs without even running the RSNM, we have the following proposition.

**Proposition 1** (degree distribution estimation). *Suppose that every node will generate an outgoing ripple; for example, Condition 3 is satisfied. For node  $i$ , let  $L_{DD}(i)$  be the value of*

$$L_{DD}(i) = \frac{\eta \alpha_L \beta_R(i)}{2\pi \beta_{L,U}}. \quad (15)$$

*Then, in the graph generated by the RSNM, the degree of node  $i$ ,  $D_N(i)$ , is not smaller than the number of nodes to which the direct distance  $L_N(i, j)$  is not larger than  $L_{DD}(i)$ .*

## 4. Genetic Algorithm (GA) to Optimize Network Topology

**4.1. Optimize Network Topology by Evolving Model Parameters.** The optimization of network topology has long been a challenging task [18], as it is widely acknowledged as an NP-hard problem. As large-scale parallel stochastic search and optimization algorithms, genetic algorithms (GAs), if properly designed, have the capability of producing high quality solutions to NP-hard problems in an acceptable period of time [19]. Actually, GAs have already been used to optimize some network structures, for example, the topology optimization of CCS7 network [20], MPLS network [21] and airline route networks [22]. However, in such studies

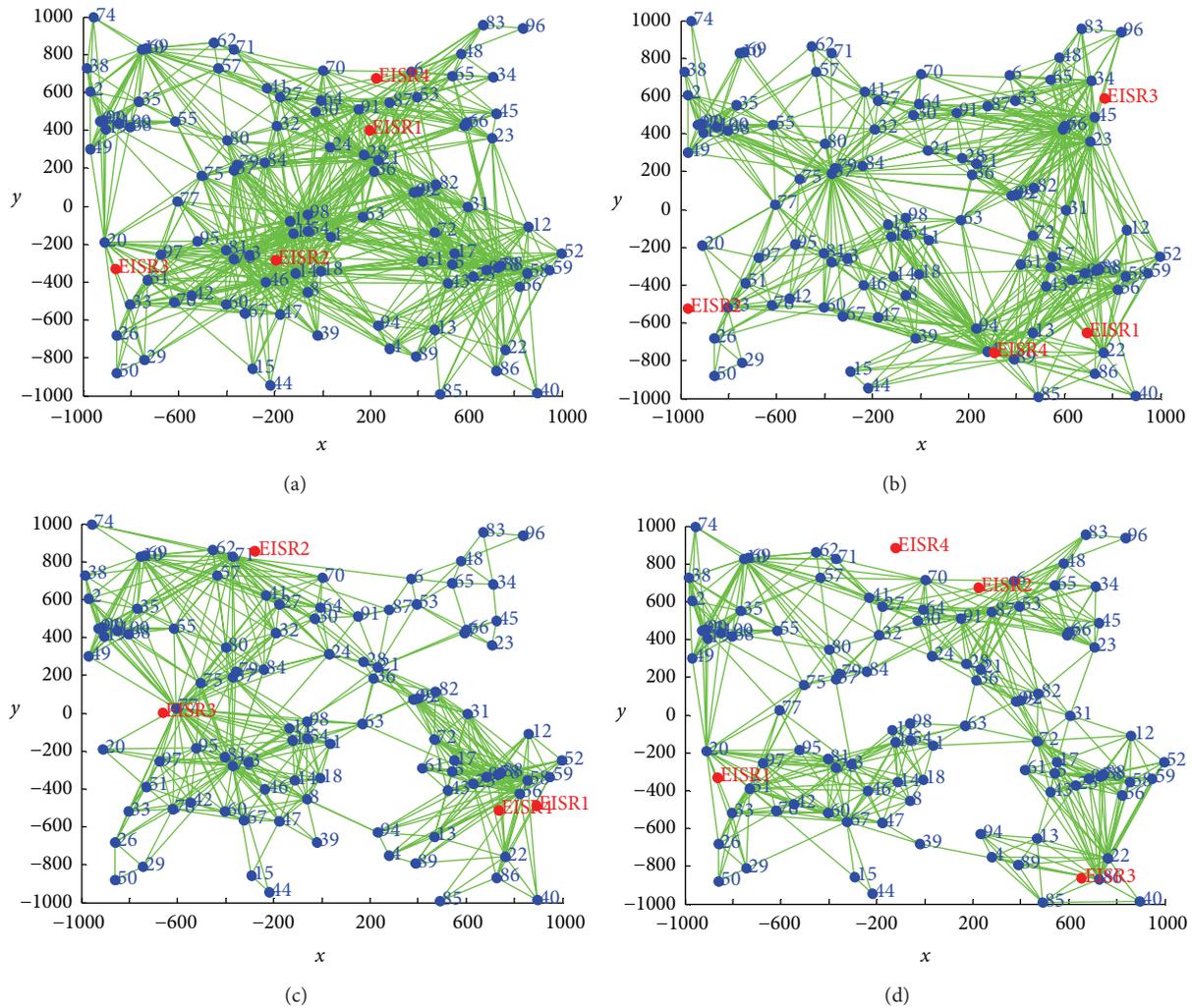


FIGURE 2: An illustration regarding how the distribution of EISRs affects network topology.

on network topology optimization, search is carried out by directly adding new links or removing existing links between nodes. The chromosome structure in such GAs is based on permutation representation of existing links. As is well known, permutation representation based GAs are often confronted with feasibility problems, which means random evolutionary operations, such as mutation and crossover, may generate some chromosomes whose associated solutions are invalid or infeasible against the underlying physical meaning of a real-world solution. For example, in the case of airline route network optimization, permutation representation based GA might establish a link between two very close cities, which is economically impracticable [22].

The RSNM makes it possible to discard permutation representation based GA, and to employ the very original binary representation based GA for network topology optimization. The original GA was developed with binary representation of value-based solutions (see [1, 2]), and it was free of feasibility problems. From the modeling process proposed in Section 3.2, one can see that the network topology is largely determined by the RSRPs. Actually, every given set of values for the RSRPs will result in a unique topology, and given

different values for the parameters, different topologies will be generated. Therefore, by evolving the values of RSRPs, one may find an optimal network topology in terms of particular considerations. Obviously, the very original binary representation based GA can be applied straightforwardly to evolve the RSRPs, just like in normal parameter optimization problems [2].

For the sake of simplicity, Figure 2 gives four examples about how the values of the RSRPs affect the output topology of the EISR-based RSNM. In the four networks in Figure 2, all other RSRPs have the same values, but the locations of the four EISRs are different. As a result, there are four different topologies. It should be pointed out that the feature that every given set of values for the model parameters will lead to a unique topology is distinct from other models of random networks, where the model parameters (e.g., the connection probability in the classic random graph theory and the preferential attachment in the theory of evolving networks) cannot uniquely determine network topology [5, 6].

One may argue that changing the values of RSRPs might never lead to every possible network topology. In other words, a certain topology might exist and no matter how

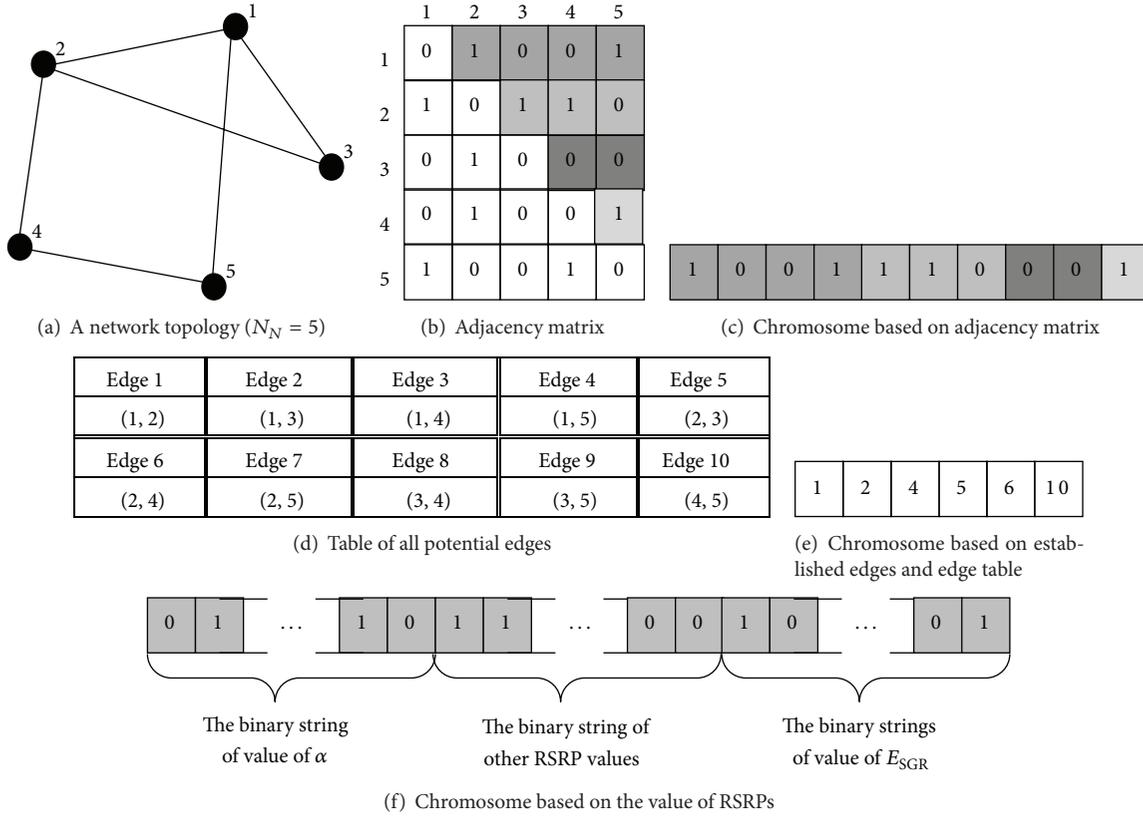


FIGURE 3: Some chromosome structures used in network topology optimization.

the RSRPs are tuned, the RSNM cannot generate it to the nature of the ripple-spreading process. Although this is true on the one hand, on the other, this feature of RSNM is not necessarily a disadvantage. In many applications of GAs, heuristic rules usually play a crucial role in delivering a good performance because they may filter out many infeasible or bad quality solutions and therefore narrow down the search space for GAs. The nature of the ripple-spreading process may do the same good thing for network topology optimization. Suppose that the development of a real-world network system largely involves some ripple-spreading effect (e.g., in the cases of disaster spreading and epidemic dynamics). Then, any network topology that cannot reflect such ripple-spreading effect is hardly realistic from the real-world system point of view and therefore should be considered as a bad solution to topology optimization. This means, if the ripple-spreading effect is concerned, then most network topologies missed out by the RSNM are possibly bad solutions. As demonstrated in [23], a ripple-spreading process can focus the search on quality solutions to the aircraft sequencing problem. Similarly, the RSNM can focus on more realistic topologies, and therefore, evolving the RSRPs could still be an effective strategy to find a good or even optimal network topology. Further study based on some particular real-world systems are needed, which is however beyond the scope of this paper.

4.2. *GA Design.* In the practice of GA design, we usually need to determine chromosome structure, mutation, crossover, and heuristic rules. These four design steps are often highly problem dependent in a permutation representation based GA but quite simple and straightforward in a binary representation based GA, which is the case in this study as just explained in Section 4.1.

Traditionally, to apply GA to optimize network topology, a straightforward chromosome structure is based on the adjacent matrix or a vector of established edges, just as illustrated in Figures 3(b)–3(e), and such a chromosome structure apparently belongs to permutation representation. Differently, with the RSNM, we can optimize network topology by evolving the values of RSRPs, and therefore, we can set the chromosome structure as a value based binary representation. Basically, the value of each RSRP can be represented as a binary string. We stick together all binary strings of a set of values for RSRPs, and then we get the associated chromosome (see Figure 3(f), e.g.). In the RSNM, a different node  $i$  may have different  $\alpha(i)$ ,  $\beta_R(i)$ , and  $\beta_L(i)$ , that is, different amplifying factors, threshold to generate a responding ripple, and threshold to establish a link, and a different SGR may have a different  $E_{SGR}(i)$ . In this study, we assume that  $\alpha(i)$  has a normal distribution within the range  $[\alpha_L, \alpha_U]$ ,  $\beta_R(i)$  within  $[\beta_{R,L}, \beta_{R,U}]$ ,  $\beta_L(i)$  within  $[\beta_{L,L}, \beta_{L,U}]$ , and  $E_{SGR}(i)$  within  $[E_{SGR,L}, E_{SGR,U}]$ . Therefore, we need to

use GA to tune the boundary values of the above 4 ranges, in other words, to use GA to optimize  $\alpha_L$ ,  $\alpha_U$ ,  $\beta_{R,L}$ ,  $\beta_{R,U}$ ,  $\beta_{L,L}$ ,  $\beta_{R,U}$ ,  $E_{SGR,L}$ , and  $E_{SGR,U}$ . As a result, a chromosome should be the integration of the binary strings of the above 8 boundary values. If we want to see what is the best topology, the RSNM could output with all nodes having the same value for certain RSRP, for example, demanding all nodes to have the same  $\alpha(i)$ , then a chromosome will become simpler because  $\alpha_L = \alpha_U$ . Furthermore, we may predetermine and fix certain RSRPs and use GA only to tune other RSRPs because, according to the conditions given in Section 3.3, one can see that it is not the absolute value of RSRPs, but the relative ratio between RSRPs that largely determine the output topology of the RSNM. This will further simplify the chromosome structure.

Mutation aims to increase the diversity of chromosomes, so that the GA can explore the solution space as widely as possible in order to stand a better chance to hit a global optimum. For a binary representation based GA, the mutation operator randomly selects some genes in a chromosome with a certain probability and reverses their values, that is, changing a binary bit from 1 to 0, or from 0 to 1.

Crossover is used to identify, inherit, and protect common genes shared by fit chromosomes, and at the same time, to recombine noncommon genes searching for new solutions. Crossover is crucial for GAs to quickly locate optima or suboptima. The simplest crossover is one-point crossover, where a split point is chosen randomly, each of the two parents split at the chosen point into two pieces, and piece 1 (or piece 2) from parent 1 is combined with piece 2 (or piece 1) from parent 2 to generate offspring. In this paper, we choose uniform crossover, mainly because it is more powerful in terms of exploiting all possibilities of recombining noncommon genes [24]. In uniform crossover, each gene in the offspring inherits the same gene from either parent 1 or parent 2 at a half-to-half chance. Actually, from Figure 4, one can see that uniform crossover is the ultimate multipoint crossover.

Problem-specific heuristic rules are used in GA to filter out bad/invalid/infeasible solutions and therefore to accelerate the converge speed. They are usually more useful and more important for permutation representation based GA rather than binary representation based GA. In the RSNM, although the ripple-spreading process is supposed to rule out most unrealistic solutions, we can still introduce some heuristic rules based on those conditions given in Section 3.3. For example, when initializing a chromosome, the random setup of RSRP values may more or less refer to those ratio relationships between the RSRPs, in order to avoid such cases where many nodes are not connected or all nodes are completely connected.

In the GA, the population of chromosomes evolves based on their fitness through generations of evolutionary operations: selection, mutation, and crossover. Since the goal in this study is to tune the RSNM to generate certain desirable topologies, the fitness of a chromosome is basically measured by network properties of the topology associated with the chromosome. It is known that small-world and scale-free topologies are mainly recognized by their node degree

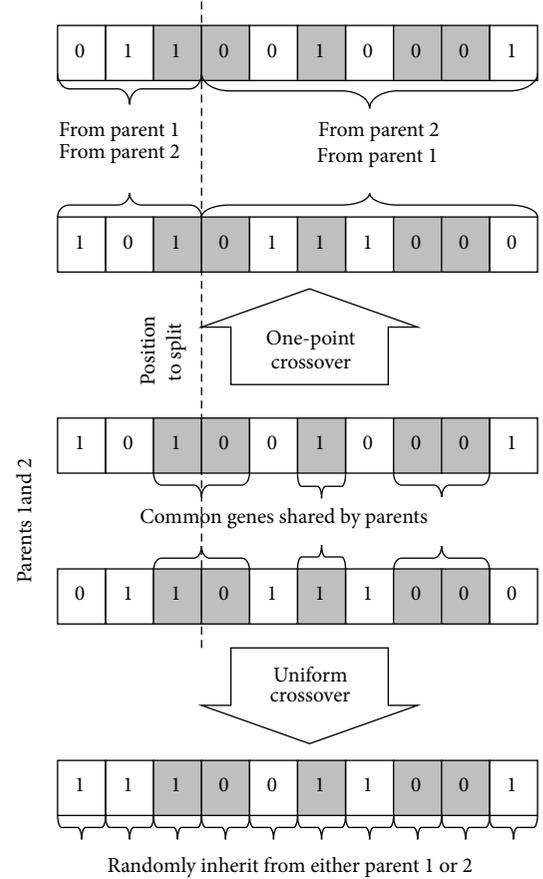
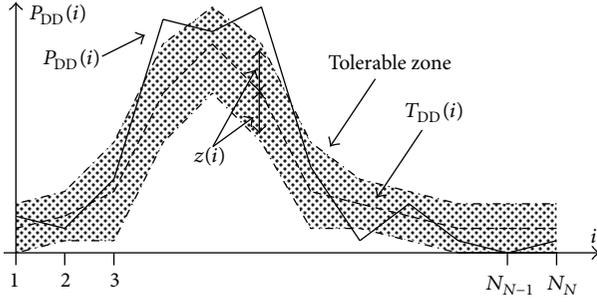


FIGURE 4: One-point crossover versus uniform crossover.

distribution pattern, average path length (APL) between connected nodes, and the clustering coefficient (CC). Basically, a small-world network topology has a Poisson distribution, small APL, and large CC, while a scale-free network topology has a power law tail-off distribution, even smaller APL, and larger CC. To apply a GA to tune the RSNM to generate small-world and scale-free topologies, we first need to use some well-known network models to generate target network topologies. In this study, we use the Watts-Strogatz model in [25] to generate small-world network topologies and the Barabási and Albert model in [17] to generate scale-free topologies. Since these models are random models, every run of such models will generate a different topology but the associated network properties are similar between different runs. Therefore, for a given model parameter value, we need to run the models many times, each time we will get a different topology, and then we calculate the average network properties of all different topologies generated. We use such average network properties as targets to tune the RSNM. Before we run the GA, suppose that we have run the one of the models and then got the following average network properties as targets:  $T_{APL}$ ,  $T_{CC}$ , and  $T_{DD}$ , that is, target APL, target CC, and target degree distribution. Please note that  $T_{APL}$  and  $T_{CC}$  are scalars, while  $T_{DD}$  is a vector which has  $N_N$  entries, and entry  $T_{DD}(i)$  is the percentage of nodes whose degree is  $(i-1)$ ,

FIGURE 5:  $P_{DD}$ ,  $T_{DD}$ , and tolerable zone.

$i = 1, \dots, N_N$ . With  $T_{APL}$ ,  $T_{CC}$ , and  $T_{DD}$ , we can calculate the fitness of a chromosome as follows in this study:

$$f = \frac{w_1}{1 + |P_{APL} - T_{APL}|} + \frac{w_2}{1 + |P_{CC} - T_{CC}|} + \frac{w_3}{1 + E_{DD}}, \quad (16)$$

where  $w_i > 0$ ,  $i = 1, 2, 3$ , are weights which determine the contribution of each network property to the fitness,

$$E_{DD} = \sum_{i=1}^{N_N} (\max(z(i), |P_{DD}(i) - T_{DD}(i)|) - z(i))^2, \quad (17)$$

$P_{APL}$ ,  $P_{CC}$ , and  $P_{DD}$  are the network properties of the topology associated with the chromosome, and  $z(i)$  defines a tolerable zone for the gap between  $P_{DD}$  and  $T_{DD}$ .  $E_{DD}$  indicates how similar the degree distribution associated with the chromosome is to the target one (see Figure 5 for illustration). From the definition of fitness function in (16), one can see that if the topology associated with the chromosome has similar network properties to those of target topology, then the chromosome will have a large fitness.

It should be pointed out that, if we assume that all nodes have the same  $\alpha(i)$ ,  $\beta_R(i)$ , and  $\beta_L(i)$ , and all SGR have the same  $E_{SGR}(i)$ , then we have  $\alpha_L = \alpha_U$ ,  $\beta_{R,L} = \beta_{R,U}$ ,  $\beta_{L,L} = \beta_{R,U}$ , and  $E_{SGR,L} = E_{SGR,U}$ , and a chromosome will output a unique topology, whose network properties can be used straightforwardly to calculate the fitness according to (16) and (17). However, in most cases when we tune the RSNM, we allow certain RSRP to have a normal distribution with a range, for example, allowing  $\alpha(i)$  to vary within  $[\alpha_L, \alpha_U]$ , where  $\alpha_L \neq \alpha_U$ . In such a case, although a chromosome gives values for  $\alpha_L$  and  $\alpha_U$ , the output of RSNM is not unique. Therefore, to calculate the fitness of a chromosome, we need to run the RSNM for many times and then use the average network properties of all generated topologies.

## 5. Simulation Results

**5.1. Simulation Setup.** In this section, we aim to investigate, when and how the RSNM may generate small-world and scale-free network topologies. To this end, firstly, we need some models which can well generate such two categories of topologies, and as indicated in Section 4.2, we used established models from the literature [17, 25]. From the modeling

process in Section 3.2, one can easily see that, even if all RSRPs are the same, the spatial distribution of nodes can largely affect the final output topology. Therefore, in this simulation, we consider 3 different types of node distribution: circle distribution, grid distribution, and random distribution. In circle distribution, all nodes are evenly distributed along a circle whose radius is 1000. In grid distribution, all nodes are evenly and regularly arranged in a rectangular area defined by two points:  $(-1000, -1000)$  and  $(1000, 1000)$ . A random distribution is generated by introducing random disturbances to the location of nodes in a grid distribution.

The Watts-Strogatz model [25] can generate lattice topology, small-world network, and random graph by increasing the probability for randomly rewiring lattice edges, denoted as  $P_{RRLE}$  hereafter, from 0 to 1. In this simulation, we choose  $P_{RRLE} = 0, 0.08$  and 1, respectively. Then, combining a different node distribution and  $P_{RRLE}$  values, we have nine test scenarios for this model denoted from WS1 to WS9 as shown in Table 1.

In the Barabási and Albert model [17], the probability for preferential attachment (denoted by  $P_{PA}$  hereafter) plays a crucial role in generating scale-free topology. In this simulation, for node  $i$ , the  $P_{PA}$  is formulated as

$$P_{PA}(i) = \frac{D_N(i)^\tau}{\sum_{j=1}^{N_N} D_N(j)^\tau}. \quad (18)$$

Basically, the coefficient  $\tau$  determines the power law of the resulted scale-free network. Obviously, if  $\tau = 0$ , then one has an almost random graph, whilst if  $\tau$  is too large, say,  $\tau \geq 2$ , then a winner-takes-all network topology is likely to emerge. Here, we will use nine Barabási and Albert test scenarios, denoted as from BA1 to BA9, with different  $\tau$  and node distribution, as shown in Table 2.

Different sets of values for RSRPs in the RSNM might lead to similar topologies. Therefore, in the simulation, by referring to the key network properties of the associated target topology, we use the reported GA to generate six different sets of RSRP values in order to compare with the WS model, and 10 different sets of RSRP values for comparing with the BA model. Each set of RSRP values determines a specific RSNM scenario, and these 16 RSNM test scenarios are denoted as from RSNM1 to RSNM16, as listed in Tables 1 and 2. To tune the RSRP values in each RSNM test scenario, the reported GA has a population of 100, a maximal generation of 200, a mutation probability of 0.1, and a crossover probability of 0.4. For the fitness function defined by (16) and (17), we set  $w_1 = 2$ ,  $w_2 = 1$ ,  $w_3 = 5$ , and  $z(i) = 0.02$ . For the set of RSRP values given by a chromosome, the RSNM is run to generate 20 network topologies, and then the average values of the associated key network properties are used to calculate the fitness of that chromosome. The populations of chromosomes compete and evolve through generations according to their fitness. Then, the fittest chromosome in the last generation of a GA run determines the set of RSRP values for the associated RSNM test scenario. The RSRP values tuned by the reported GA for each of the 16 RSNM test scenarios are given in Tables 1 and 2. In some test scenarios, some RSRPs are assumed to be the same for all nodes, that is, have the same lower bound

TABLE 1: Comparative results: to generate small-world network topology.

NDT		Setup of model parameters	APL	ADEE	CC	ASSO
WS1	Circle dist.	$P_{RRLE} = 0$	12.8788	94.2	0.5000	0.0000
WS2	Circle dist.	$P_{RRLE} = 0.08$	4.4299	305.7	0.3117	0.0082
WS3	Circle dist.	$P_{RRLE} = 1$	3.5061	1325.0	0.0465	0.1795
WS4	Grid dist.	$P_{RRLE} = 0$	6.6667	222.2	0.0000	0.4010
WS5	Grid dist.	$P_{RRLE} = 0.08$	4.2663	365.9	0.2399	0.0636
WS6	Grid dist.	$P_{RRLE} = 1$	3.5262	1024.0	0.0549	0.2630
WS7	Random	$P_{RRLE} = 0$	6.2159	239.9	0.3487	0.0557
WS8	Random	$P_{RRLE} = 0.08$	4.3079	362.3	0.2468	0.1269
WS9	Random	$P_{RRLE} = 1$	3.5132	1079.9	0.0758	0.2356
RSNM1	Circle dist.	$\beta_R = 5.1, \beta_L = 3.1, \alpha = 440.5, N_{SGR} = 10, \text{ and } E_{SGR}(i) \in [2489.6, 3507.7]$	12.8788	94.2	0.5000	0.0000
RSNM2	Circle dist.	$\beta_R(i) \in [0.6, 1.2], \beta_L(i) \in [0.1, 0.8], \alpha(i) \in [482.4, 541.7], N_{SGR} = 10, \text{ and } E_{SGR}(i) \in [313.5, 650.0]$	8.1432	151.9	0.4836	0.0598
RSBM3	Grid dist.	$\beta_R = 4.5, \beta_L = 4.1, \alpha = 1413.7, N_{SGR} = 10, \text{ and } E_{SGR}(i) \in [3927.3, 17181.8]$	6.6667	222.2	0.0000	0.4010
RSNM4	Grid dist.	$\beta_R(i) \in [4.5, 6.2], \beta_L(i) \in [2.3, 7.5], \alpha(i) \in [1507.3, 1619.1], N_{SGR} = 10, \text{ and } E_{SGR}(i) \in [3901.2, 17215.5]$	4.9061	288.2	0.3396	0.0962
RSBM5	Random	$\beta_R = 4.5, \beta_L = 4.5, \alpha = 1319.5, N_{SGR} = 10, \text{ and } E_{SGR}(i) \in [3915.3, 17198.1]$	5.4315	240.2	0.4348	0.1031
RSNM6	Random	$\beta_R(i) \in [4.5, 6.1], \beta_L(i) \in [1.9, 7.0], \alpha(i) \in [1853.8, 1527.2], N_{SGR} = 10, \text{ and } E_{SGR}(i) \in [3897.9, 17301.2]$	4.0364	314.2	0.5542	0.1741

TABLE 2: Comparative results: to generate scale-free network topology.

NDT		Setup of model parameters	APL	ADEE	CC	ASSO
BA1	Circle dist.	$\tau = 0$	2.6561	1235.6	0.0600	0.0324
BA2	Circle dist.	$\tau = 1$	2.5103	1272.5	0.1339	0.1435
BA3	Circle dist.	$\tau = 2$	1.9350	1279.5	0.6875	0.4213
BA4	Grid dist.	$\tau = 0$	2.4652	1146.6	0.0725	0.0260
BA5	Grid dist.	$\tau = 1$	2.3690	1154.5	0.1434	0.1232
BA6	Grid dist.	$\tau = 2$	1.9224	1143.8	0.6492	0.3930
BA7	Random	$\tau = 0$	2.2271	1146.7	0.0953	0.0369
BA8	Random	$\tau = 1$	2.1768	1153.1	0.1549	0.1129
BA9	Random	$\tau = 2$	1.8991	1151.1	0.6214	0.3033
RSNM7	Circle dist.	$\beta_R = 5, \beta_L = 5.1, \alpha = 1036.7, N_{SGR} = 100, \text{ and } E_{SGR}(i) \in [0, 56549]$	2.3208	634.5	0.7718	0.3593
RSNM8	Grid dist.	$\beta_R = 5, \beta_L = 5.0, \alpha = 1036.7, N_{SGR} = 100, \text{ and } E_{SGR}(i) \in [0, 62832]$	2.1030	599.9	0.7392	0.3272
RSBM9	Random	$\beta_R = 5, \beta_L = 5.1, \alpha = 1036.7, N_{SGR} = 100, \text{ and } E_{SGR}(i) \in [0, 62832]$	2.1857	564.9	0.7397	0.3234
RSNM10	Circle dist.	$\beta_R = 100, \beta_L(i) \in [1.1, 15.2], \alpha = 1036.7, N_{SGR} = 100, \text{ and } E_{SGR} = 62832$	9.8945	213.1	0.5579	0.0916
RSBM11	Grid dist.	$\beta_R = 100, \beta_L(i) \in [1.0, 10.3], \alpha = 1036.7, N_{SGR} = 100, \text{ and } E_{SGR} = 10053$	2.0285	693.1	0.8448	0.3445
RSNM12	Random	$\beta_R = 100, \beta_L(i) \in [0.5, 10.2], \alpha = 1036.7, N_{SGR} = 100, \text{ and } E_{SGR} = 13100$	2.4119	591.7	0.7648	0.3947
RSNM13	Circle dist.	$\beta_R = 5, \beta_L = 5.1, \alpha(i) \in [439.8, 3141.6], N_{SGR} = 1, \text{ and } E_{SGR} = 2199.1$	5.0135	365.7	0.6619	0.0138
RSNM14	Grid dist.	$\beta_R = 5, \beta_L = 9.2, \alpha(i) \in [2585.0, 6911.5], N_{SGR} = 1, \text{ and } E_{SGR} = 9424.8$	2.8947	473.5	0.6374	0.1715
RSBM15	Random	$\beta_R = 5, \beta_L = 8.7, \alpha(i) \in [2585.0, 6911.5], N_{SGR} = 1, \text{ and } E_{SGR} = 9419.3$	2.1877	547.8	0.7099	0.1956
RSNM16	Random	$\beta_R = 5, \beta_L = 0.5, \alpha = 471.2, N_{SGR} = 1, \text{ and } E_{SGR} = 7854.0$	1.9594	902.1	0.9648	0.9524

as the upper bound, while other RSRPs are assumed to have a normal distribution within certain ranges. For example, in RSNM1,  $\beta_R$ ,  $\beta_L$ , and  $\alpha$  are the same for all nodes, while  $E_{SGR}(i)$  may vary within a range. Then, in RSNM, we use GA to tune  $\beta_R$ ,  $\beta_L$ ,  $\alpha$ , and the boundary values for  $E_{SGR}(i)$ .

For each test scenario, the associated network model runs for 100 times, and the average values of five important network properties, that is, the average path length (APL)

between connected nodes, the average distance of established edges (ADEE), the clustering coefficient (CC), the assortativity (ASSO), and the degree distribution (DD), are calculated and analyzed. Please note the difference between APL and ADEE: APL is measured as the number of intermediate nodes, while ADEE is defined as physical distance.

All models are coded and all tests are conducted in a MATLAB environment on a personal computer with

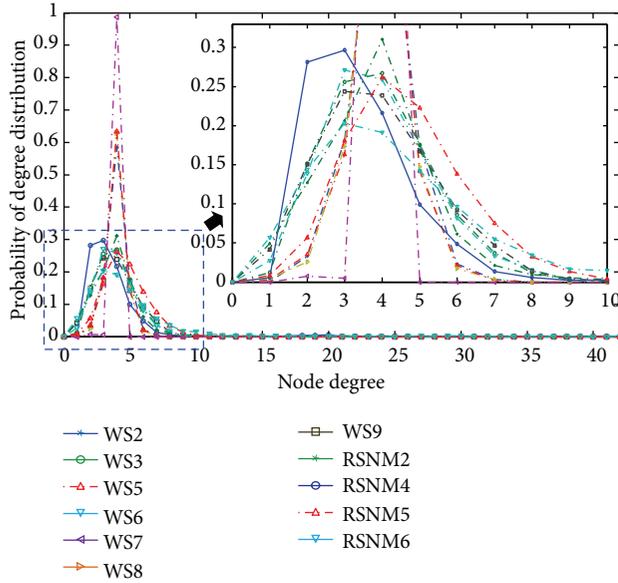


FIGURE 6: Average degree distribution of small-world networks generated by WS and RSNM.

a 2.6 GHz CPU, 4 GB memory, and the Windows 7 operating system.

**5.2. Simulation Results for the Small-World Topology.** In this subsection, we have  $N_N = 100$  nodes in each test of the simulation and then use a certain network model to establish  $N_L = 200$  edges between nodes to get a network topology. The number of SGR is always set as  $N_{SGR} = 10$  for the RSNM. The simulation results of WS1 to WS9 and RSNM1 to RSNM6 are given in Table 1 and Figure 6, and some topology examples generated during the simulation are given in Figure 7. From the table and figures, one may have the following observations.

- (i) Firstly, the data of WS1 to WS9 in Table 1 show that, compared with lattices (when  $P_{RRLE} = 0$ ) and random networks (when  $P_{RRLE} = 1$ ), a distinctive feature of small-world networks, as analyzed in [12], is that they have relatively smaller APL (compared with the associated lattices) and larger CC (compared with the associated random networks). Therefore, the first aim of this simulation is to test whether the reported GA can tune RSRPs properly to make the RSNM to generate networks with the same feature.
- (ii) From the data of RSNM2, RSNM4, RSNM5, and RSNM6, one can see that the RSNM is indeed capable of generating topologies with the above small-world network feature.
- (iii) Compared with the small-world networks generated by the WS, Table 1 also shows that the small-world networks generated by the RSNM usually have larger APL and CC (e.g., compare WS2 with RSNM2, WS5 with RSNM4, and WS8 with RSNM5 and RSNM6). This is understandable, because the

process of rewiring lattice edges in the WS is purely random, regardless of spatial distance between nodes. Due to such a random process, the WS has more edges established between spatially far away nodes, which help to reduce APL more significantly than the RSNM. Random rewiring lattice edges also lead to smaller CC when compared with the RSNM because the ripple-spreading process in the RSNM is more likely to establish edges within neighbourhood, which usually causes large CC.

- (iv) Also because of the difference between the random rewiring process in the WS and the ripple-spreading process in the RSNM, the small-world networks of RSNM usually have smaller ADEE. An interesting finding is that, when multiplying APL with ADEE, the small-world networks generated by the RSNM generally have the smaller travelling distance between two randomly chosen nodes than those of the WS actually than all networks generated by the WS including lattices and random networks.
- (v) When comparing the data of WS1 and WS4, that is, two lattice cases, with those of RSNM1 and RSNM3, one can see that the RSNM can also generate exactly the same lattices if the RSRPs are set properly. The WS initializes lattice with the spatial information of neighborhood; this explains when the RSNM can also generate lattices (the RSNM is particularly good at using spatial information to generate networks).
- (vi) Figure 6 compares the degree distribution of different models, except WS1, WS4, RSNM1, and RSNM3 (they are lattices and their degree distribution is simply a singleton). Basically, in the WS, the smaller the value of  $P_{RRLE}$ , the sharper the degree distribution, in other words, the less diversified the node degree. Overall, no matter whether we consider lattice, small-world topology, or random network, the degree distribution is similar to a Poisson distribution. In general, the degree distribution of small-world networks generated by the RSNM is more diversified than that of WS small-world networks. This is probably because the ripple-spreading process is more complicated than the random rewiring process, and therefore the node degree is less precisely managed in the RSNM than in the WS. Actually, the degree distribution of RSNM small-world networks is more like that of WS random networks (WS3, WS6, and WS9).
- (vii) Figure 7 gives some examples of small world networks generated by the WS and the RSNM (for the sake of comparison, Figure 7(d) is a random network, and Figure 7(h) is a network where edges are only established between neighborhood nodes). From Figure 7, one can see clearly that long-distance jump edges in the WS are completely random, whilst in the RSNM, they are mainly established around a few hub-like nodes. This feature, along with the degree distribution of RSNM4 in Figure 6, which is the least Poisson-distribution-like, may imply that the RSNM is more

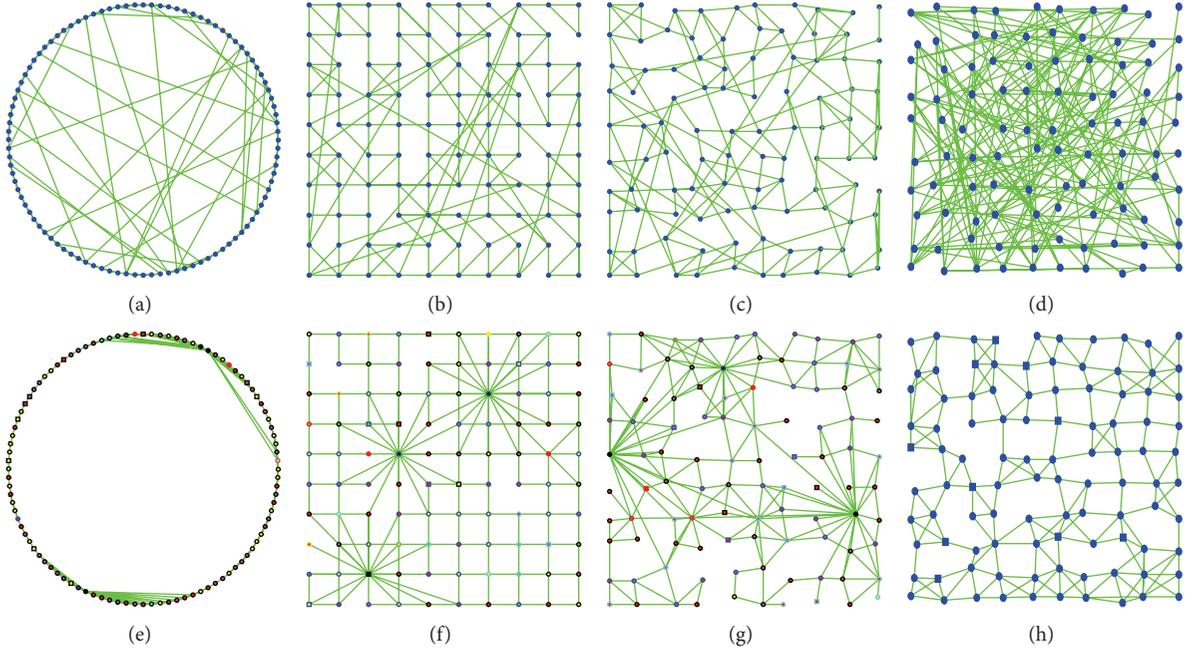


FIGURE 7: Examples of small-world networks generated by WS and RSNM: (a) WS2; (b) WS5; (c) WS8; (d) WS9; (e) RSNM2; (f) RSNM4; (g) RSNM6; (h) RSNM5.

likely a natural scale-free network model. This will be further investigated in the next section.

- (viii) In real-world networks, a node may not connect to its neighborhood nodes but some far away nodes. Figures 7(e)–7(g) clearly show that this is achievable in the RSNM, for instance, when closer nodes have large  $\beta_L$ , while some far away nodes have small enough  $\beta_L$ . However, the structure of connecting far away nodes rather than closer ones does not emerge as often in the RSNM as in the WS.
- (ix) The random network Figure 7(d) is generated by the WS. In this study, we have realized that it is very difficult to tune the RSRPs in order to generate purely random networks. Basically, because spatial distance plays a crucial role in the RSNM, we assume that it is impossible for the RSNM to generate purely random networks like Figure 7(d).

**5.3. Simulation Results about Scale-Free Topology.** In this subsection, we again have  $N_N = 100$  nodes in each test of the simulation, the number of established edges  $N_L$  in the RSNM may vary depending on how fast ripple energy decays, and then the comparative BA model is required to generate a network with the same edge number. The number of SGR is always set as  $N_{SGR} = 100$  for RSNM7 to RSNM12 and  $N_{SGR} = 1$  for RSNM13 to RSNM16. As discussed in Section 3.2, it is the ratio relationships between the RSRPs that largely determine the output topology; therefore, in this subsection, we preset  $\beta_R = 5$  for RSNM7 to RSNM9 and RSNM13 to RSNM16 and preset  $\beta_R = 100$  for RSNM10 to RSNM12. The simulation results of BA1 to BA9 and RSNM7 to RSNM16 are given in

Table 2 and Figures 8 and 9, from which one may make the following observations.

- (i) Firstly, let us analyze the networks generated by the classical scale-free model. From the data of BA1 to BA9 in Table 2, one can see clearly that, as  $\tau$  goes up from 0 to 2, the APL decreases whilst both the CC and the ASSO increases. Compared with lattices, small-world networks and random networks, such as WS1 to WS9 in Section 5.2, and scale-free networks in general have much smaller APL and larger CC and ASSO. Hub nodes in scale-free networks contribute a lot to smaller APL. In a winner-takes-all network, for example, when  $\tau \geq 2$ , a major hub node may connect to all other nodes, which means that the APL is less than 2. The large CC results from the fact that, in a scale-free network, two connected nonhub nodes are likely to connect to the same hub node, and therefore a three-node cluster is formed. In a winner-takes-all network, any two non-hub nodes connect to the hub, so the associated CC is very large. Since the ASSO is an index to indicate the preference for a network node to attach to others that are similar or different in any way [26], it of course goes up with the number of preferential attachments, which is determined by  $\tau$ .
- (ii) With properly tuned RSRPs by the reported GA, Table 2 shows that the RSNM can generate scale-free networks indeed, as the RSNM networks share the same features as the BA scale-free networks, that is, small APL and large CC and ASSO. It should be noted that, in the BA experiments, the only parameter  $\tau$  is increased gradually in order to reveal the change

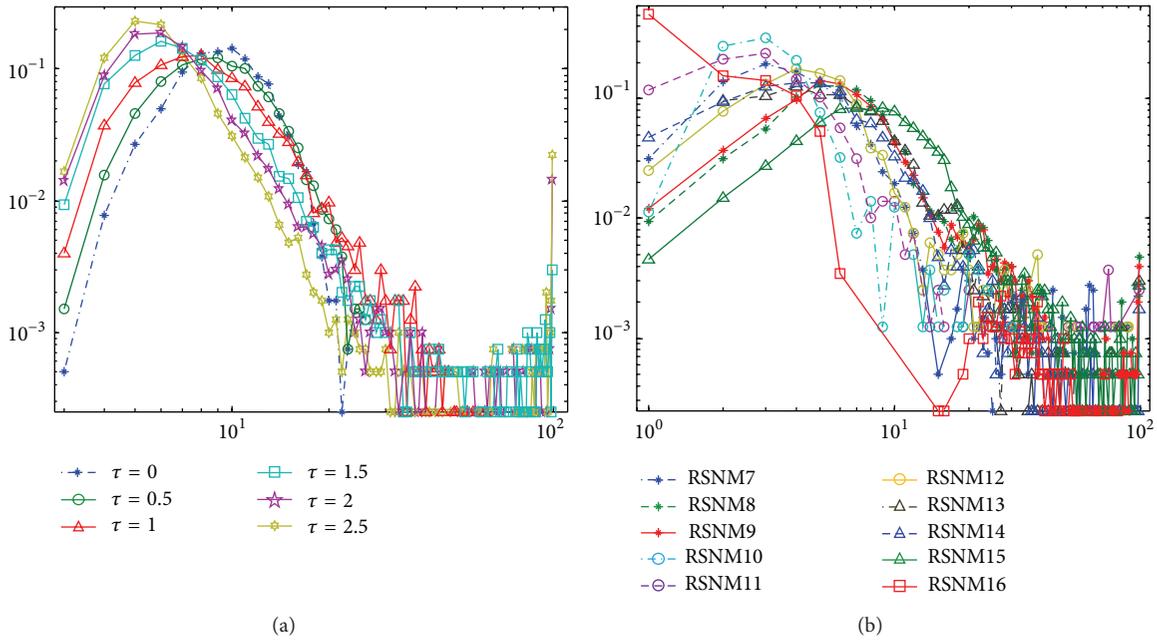


FIGURE 8: Average degree distribution of scale-free networks generated by BA and RSNM: (a) BA with random node distribution and different values for  $\tau$ ; (b) RSNMs.

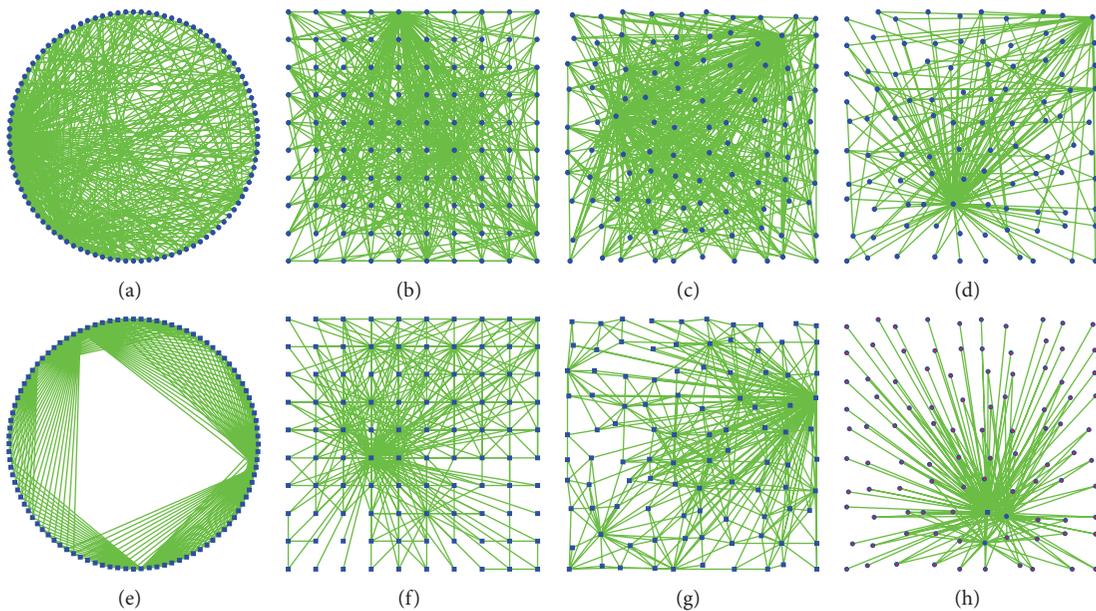


FIGURE 9: Examples of scale-free networks generated by BA and RSNM: (a) BA2; (b) BA5; (c) BA8; (d) BA9; (e) RSNM7; (f) RSNM11; (g) RSNM15; (h) RSNM16.

trend in network properties, while in the RSNM experiments, the changes in RSRPs are not continuous or even associated, and they are just tuned with purpose by referring to the BA scale-free networks.

(iii) Table 2 also shows that, in general, the RSNM networks have much smaller ASSO than the associated BA networks. This is reasonable because the BA model does not consider spatial distance at all when

establishing an edge. By multiplying the APL and ASSO, one can see again that the RSNM networks are more efficient in terms of overall travelling distance between two randomly chosen nodes. As is well known, under restricted resources, scale-free networks are highly appreciated because they are efficient in terms of overall travelling distance. For instance, given limited aircraft and staff, most airline companies now operate on hub-and-spoke airline route

networks, which is an engineering version of scale-free network [27]. Compared with the BA, the RSNM can make more of this advantage of scale-free works.

- (iv) Figure 8(a) shows how the degree distribution changes when  $\tau$  increases from 0 to 2.5 in the BA. Basically, for a scale-free network, the main part of the right side slope of the degree distribution appears linear in a log-log plot, and the slope becomes sharper as  $\tau$  increases. Figure 8(b) gives the degree distribution of RSNM7 to RSNM16, from which one can clearly see they have approximately linear right side slope with a different sharpness. This further confirms that the RSNM can generate scale-free networks, given that the RSRPs are properly tuned.
- (v) Figure 9 gives some examples of scale-free networks generated by the BA and the RSNM. Apparently, the BA scale-free networks are more spatially chaotic, whilst the RSNM networks are much better organized in terms of spatial view.
- (vi) One may notice that, in Table 2, the APL of RSNM10 and RSNM13 is much larger than others. This is mainly because of the circle node distribution used. In a scale-free network, it often needs to pass a hub node to travel between two nodes. In the RSNM, even a hub node tends to establish edges locally. Therefore, in the case of circle node distribution, sometimes it may need to pass through quite a few hub nodes around the circle to travel between two nodes.

## 6. Conclusions and Future Work

This paper is concerned with how to apply a genetic algorithm (GA) to tune the parameters of ripple-spreading network model (RSNM), in order to generate small-world and scale-free network topologies. As a newly reported complex network model, the RSNM may well describe the development of many real-world network systems where the ripple-spreading effect of some local events' influence plays an important role. However, a demerit is that those ripple-spreading related parameters (RSRPs) of RSNM have no obvious or simple relationship with the network properties of output topology. In particular, since small-world and scale-free topologies are two of the most important features of many real-world complex network systems, it is crucial to understand, when and how the RSNM can generate these two categories of network topologies. This paper develops an effective GA to tune the RSRPs in the RSNM, and the simulation shows that the RSNM has a great flexibility in generating various network topologies. The reported work further verifies the theoretical potential of RSNM. Therefore, future work should pay more attention to real applications of the RSNM (such as in the study of disaster management and epidemic dynamics), and the GA developed in this paper can then be easily extended to tune model parameters, which will play an important role in simulating a real-world system. For example, when applying the RSNM to study a specific real-world network system, comparison should be conducted between the reported GA

and existing relevant methods for that specific system in terms of time performance and scalability.

## Acknowledgments

This work was supported in part by the China "973" Project under Grant 2012CB955404, the Project Grant 2012-RC-02 from Beijing Normal University, China, and the Seventh Framework Programme (FP7) of the European Union under Grant PIOF-GA-2011-299725.

## References

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, New York, NY, USA, 3rd edition, 2010.
- [3] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.
- [4] X.-B. Hu, M. Wang, M. S. Leeson, E. L. Hines, and E. Di Paolo, "Deterministic ripple-spreading model for complex networks," *Physical Review E*, vol. 83, no. 4, Article ID 046123, 14 pages, 2011.
- [5] P. Erdős and A. Rényi, "On random graphs. I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.
- [6] F. Liljeros, C. R. Edling, L. A. Nunes Amaral, H. E. Stanley, and Y. Åberg, "Social networks: the web of human sexual contacts," *Nature*, vol. 411, no. 6840, pp. 907–908, 2001.
- [7] I. Frommer and G. Pundoor, "Small-worlds: a review of recent books," *Networks*, vol. 41, no. 3, pp. 174–180, 2003.
- [8] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pp. 171–180, ACM, New York, NY, USA, 2000.
- [9] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, Article ID 025102(R), 4 pages, 2001.
- [10] L. Barnett, E. Di Paolo, and S. Bullock, "Spatially embedded random networks," *Physical Review E*, vol. 76, no. 5, Article ID 056115, 18 pages, 2007.
- [11] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.
- [12] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.
- [13] M. Kochen, Ed., *The Small World*, Ablex, Norwood, NJ, USA, 1989.
- [14] R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.
- [15] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *Computer Communication Review*, vol. 29, p. 251, 1999.
- [16] H. Jeong, B. Tombor, R. Albert, Z. N. Oltval, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.
- [17] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

- [18] M. Minoux, "Network synthesis and optimum network design problems: models, solution methods and applications," *Networks*, vol. 19, no. 3, pp. 313–360, 1989.
- [19] A. E. Eiben and M. Schoenauer, "Evolutionary computing," *Information Processing Letters*, vol. 82, no. 1, pp. 1–6, 2002.
- [20] Z.-H. Xin and H.-J. Zhang, "Neural network and genetic algorithms for topology optimization of the CCS7 network," *International Transactions in Operational Research*, vol. 9, no. 4, pp. 427–436, 2002.
- [21] E.-S. M. El-Alfy, "MPLS network topology design using genetic algorithms," in *Proceedings of IEEE International Conference on Computer Systems and Applications*, pp. 1059–1065, March 2006.
- [22] H. Liu, X.-B. Hu, S. Yang, K. Zhang, and E. Di Paolo, "Application of complex network theory and genetic algorithm in airline route networks," *Transportation Research Record*, vol. 2214, pp. 50–58, 2011.
- [23] X.-B. Hu and E. A. di Paolo, "A ripple-spreading genetic algorithm for the aircraft sequencing problem," *Evolutionary Computation*, vol. 19, no. 1, pp. 77–106, 2011.
- [24] J. Page, P. Poli, and W. B. Langdon, "Smooth uniform crossover with smooth point mutation in genetic programming: a preliminary study," in *Proceedings of the 2nd European Workshop on Genetic Programming (EuroGP '99)*, Göteborg, Sweden, May 1999.
- [25] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [26] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, no. 20, Article ID 208701, 4 pages, 2002.
- [27] G. Burghouwt, J. Hakfoort, and J. R. van Eck, "The spatial configuration of airline networks in Europe," *Journal of Air Transport Management*, vol. 9, no. 5, pp. 309–323, 2003.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

