Ripple-Spreading Model and Genetic Algorithm for Random Complex Networks: Preliminary Study

X. B. Hu, E. Di Paolo and L. Barnett

***Abstract*—Recently complex network theory has been broadly applied in various domains. How to effectively and efficiently optimize the topology of complex networks remains largely an unsolved fundamental question. When applied to the network topology optimization, Genetic Algorithms (GAs) are often confronted with permutation representation, memory-inefficiency and stochastic modeling problems, as well as difficulties in the design of problem-specific evolutionary operators. This paper, inspired by the natural ripple spreading phenomenon, reports a deterministic model of random complex networks. Unlike existing stochastic models, the topology of a random network can be thoroughly determined by some ripple-spreading related parameters in the new model. Therefore, the network topology can be improved by optimize these ripple-spreading related parameters. As a result, no purpose-designed GA is required, but a very basic binary GA, compatible to all classic evolutionary operators, can be applied in a straightforward way. Preliminary simulation results demonstrate the potential of the proposed ripple-spreading model and GA for the topology optimization of random complex networks.**

I. Introduction

COMPLEX networks, i.e. 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 [1], [2], [3]. The study of networks can be historically traced back to the 18th century when graph theory was born, while complex network theory has just developed since the last decade, which is partially a result of the accelerated developing progress of computer technologies. Modern powerful computers make it possible to model and analyze complex networks, which can easily compose of thousands, millions or even billions of nodes and links. Complex network theory provides a mathematical platform to study most real networks which the classic graph theory can not handle. Many new concepts, measures and algorithms have been defined and developed to characterize the topology of real networks, and to cover the novel topics and problems in real networks. Complex network theory has now been widely applied to study many networks in our daily life, such as our social network, Internet, WWW (World Wide Web), power grid, transportation network, communication network, food web, and even the network of protein reactions.

In the past few decades many efforts have been made to model and analyze various complex networks [1] [2]. Most network models can be classified as stochastic model, because they have a typical feature in common: a stochastic model abstracts one or a few network properties or parameters in order to capture in quantitative terms the underlying organizing principles of complex networks, and these network properties or parameters can estimate what the network looks like, but can not 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 the classic random graph theory [4], the connection probability is the core parameter of the network model. In the generalized random graph model [5], the 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 can not capture. In the theory of evolving networks [6], the parameter, so called preferential attachment, is often used to model scale-free real networks. In a recently reported spatial embedded random network model [7], the connection probability is formulated as a function of distance between nodes. In either of above network models, even if the input is fixed, i.e., 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. Except 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 the network topology optimization problem.

In this paper we will propose a novel deterministic random network model inspired by the natural ripple-spreading phenomenon. In this model, some ripple-spreading related parameters are defined as input. Unlike those stochastic models, 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 ripple-spreading parameters in order to improve the network topology in terms of concerned network properties. Further more, the proposed ripple-spreading network model can easily be extended to a semi-deterministic version and a stochastic version.

Another big advantage of this ripple-spreading network model is: it is friendly to genetic algorithms (GAs) and

This work was supported by the EPSRC Grant EP/C51632X/1.

X. B. Hu is with the Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex, Brighton, BN1 9QH, UK (phone: +44(0)1273872606; fax: +44(0)1273877873; e-mail: Xiaobing.Hu@sussex.ac.uk).

E. Di Paolo is with the Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex, Brighton, BN1 9QH, UK (e-mail: ezequiel@sussex.ac.uk).

L. Barnett is with the Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex, Brighton, BN1 9QH, UK (e-mail: lionelb@sussex.ac.uk).

compatible to all classic evolutionary operations. The network topology optimization problem is a NP-complete problem. As large-scale parallel stochastic search and optimization algorithms, GAs, if properly designed, have the capability of producing high quality solutions to the network topology optimization problem in an acceptable period of time. Actually, attempts have already been made to apply GAs to optimize some network structures, e.g., the topology optimization of CCS7 network [8], of MPLS Network [9], of airline route networks [10] and of truss [11]. In these applications of GAs, permutation representation, e.g., based on the conventional adjacent matrix, was used to record the network topology, and the chromosome structure and evolutionary operators were then designed based on the adjacent matrix. As a result, memory-inefficiency and feasibility problems often arise. Stochastic models are seldom used by GAs to optimize the network topology, because, as mentioned before, they do not offer one-to-one mapping from input parameters to output network topology. Hopefully, the ripple-spreading deterministic model of random networks proposed in this paper enables the very basic binary GAs to optimize the network topology by evolving the ripple-spreading related parameters. As will be discussed later, a GA based on the ripple-spreading model is free of memory-inefficiency and feasibility problems.

II. RIPPLE-SPREADING MODEL OF RANDOM NETWORKS

In the complex network theory, there are many network properties which can partially determine the topology of a random network. For instance, degree distribution, preferential attachment and initial attractiveness can determine the category of a network, e.g., a small-world network or a scale-free network [1], [2]. However, these network properties are seldom used by GAs to construct chromosomes in order to optimize the network topology, as they do not give unique output. In the following ripple-spreading model, we propose a few parameters which can thoroughly determine the network topology, and therefore are suitable candidates for GA to evolve.

A. Basic idea of ripple-spreading model

The basic natural ripple-spreading phenomenon is as following. Suppose a bunch of stakes are randomly distributed in a quiet pool. Then suddenly a stone is thrown into the pool, and an initial ripple is generated from the point where the stone hits the quiet water. When the ripple reaches a near stake, a new ripple is generated around the stake due to the reflection effect. Hereafter, for the sake of consistency, we call such a new ripple as a responding ripple, and the ripple which triggers the responding ripple as a stimulating ripple. As the initial stimulating ripple is spreading, more and more responding ripples are stimulated around stakes. However, since the point energy on the initial stimulating ripple decays as it spreads out, those responding ripples triggered at a late phase could hardly be noticed.

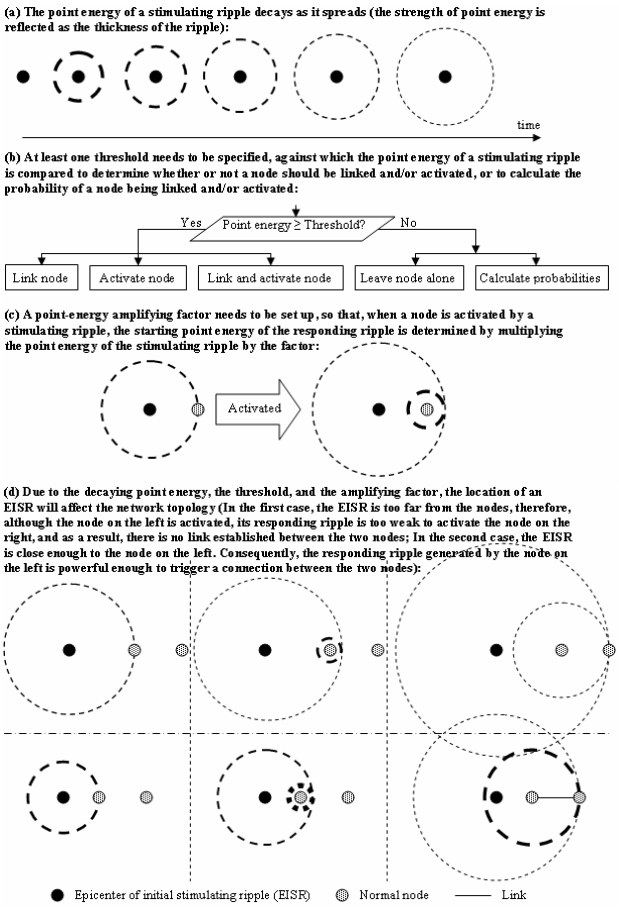


Fig.1. Mechanism of using ripple-spreading method to model networks.

Now, we replace the stakes with a set of wireless vibration sensors. Each sensor can detect and measure the point energy when a stimulating ripple reaches the sensor. If the point energy is above a preset threshold, then the sensor will generate a responding ripple, whose initial energy is a function of the point energy of the stimulating ripple, e.g., multiplying the point energy by an amplifying factor. If the stimulating ripple comes from another sensor, then the current sensor will compare the point energy against another preset threshold, once above which a permanent communication will be established between the two sensors. Obviously, the amplifying function of a sensor makes it possible for the information associated with the stone hitting the quiet pool to propagate through the whole wireless sensor network, even though the point energy of the initial stimulating ripple may decrease quickly below the threshold. Suppose each sensor can generate no more than one responding ripple, in other words, each sensor can be activated no more than once. Then, after all ripples decay, we will get a network according to those permanent communications, i.e. links, established between wireless sensors, i.e. nodes. Clearly there are some factors affecting the final network topology. For instance, how many stones hit the pool to generate initial stimulating ripples, where do they hit the pool, 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 sensor? By mathematically formulating these factors and the relationships between them, we can get a deterministic model for random networks. The mechanism of this ripple-spreading model is intuitively illustrated in Fig.1.

B. Ripple-spreading related parameters

Here we give the mathematical descriptions of those factors discussed above, which hereafter we call as ripple-spreading related parameters.

The first group of ripple-spreading related parameters are related to the epicenters of initial stimulating ripples (EISRs). Suppose N_{EISR} stones of different mass hit the pool in different points at different time instants, i.e., there are N_{EISR} EISRs, EISR i, i=1,…,N_{EISR}, has an initial point energy of $\mathrm{E}_{\mathrm{EISR}}(i)$, its coordinates are $(x_{\mathrm{EISR}}(i),y_{\mathrm{EISR}}(i))$, and it is not active until time instant $T_{\mathrm{EISR}}(i)$.

The second group of parameters associate with the nodes which are to be connected in order to generate a network. In this paper, it is assumed that the total number of nodes, N_{N}, and their locations, $(x_{\mathrm{N}}(i),y_{\mathrm{N}}(i))$, are already given and fixed. To get different topologies from this fixed set of nodes, we introduce three ripple-spreading related parameters to each node: $\alpha(i)$, $\beta_{\mathrm{R}}(i)$ and $\beta_{\mathrm{L}}(i)$, which are the amplifying factor, the threshold to generate a responding ripple, and the threshold to establish a link, for node i, i=1,…, N_{N}, respectively.

With above ripple-spreading related parameters, the proposed random network modeling process can be mathematically described as follows:

Step 1: Initialize the current time instant, i.e., t=0. Initialize the current point energy of each EISR as

$$e_{EISR}(i,t) = E_{EISR}(i),\ i=1,\ldots,N_{\mathrm{EISR}}. \quad (1)$$

Since each node has no initial energy, i.e., $E_{\mathrm{N}}(i)$=0, therefore its current point energy is

$$e_N(i,t) = E_N(i) = 0,\ i=1,\ldots,N_{\mathrm{N}}. \quad (2)$$

Assume each EISR or node has a ripple with a current radium of 0, i.e., $r_{\mathrm{EISR}}(i,t)$=0 or $r_{\mathrm{N}}(i,t)$=0.

Step 2: If the stopping criteria is not satisfied, do:

Step 2.1: Let t=t+1.

Step 2.2: Check t against $T_{\mathrm{EISR}}(i)$. If t>$T_{\mathrm{EISR}}(i)$, then update the current radium and point energy of EISR i as following

$$r_{EISR}(i,t) = r_{EISR}(i,t-1) + s, \quad (3)$$

$$e_{EISR}(i,t) = f_{Decay}(E_{EISR}(i), r_{EISR}(i,t), t) \quad (4)$$

Where s is the spreading speed of ripples, i.e., the change in the radium of a ripple during one time instant, and f_{Decay} is a function defining how the point energy decays as the ripple spreads out. A typical decaying function may be

$$f_{Decay}(E_{EISR}(i), r_{EISR}(i,t), t) = \eta \frac{E_{EISR}(i)}{2\pi r_{EISR}(i,t)} \quad (5)$$

where η is coefficient, and π is the mathematical constant. Clearly η has an important influence on the decaying speed of ripples, and will therefore affect the final network topology. As a result, η should be optimized along with other ripple-spreading related parameters.

Step 2.3: Check which new nodes are reached by the ripples of EISRs. Suppose $D_{\mathrm{EISR}}(i,j)$ is the distance between EISR i and node j. If $E_{\mathrm{N}}(j)$=0 and $D_{\mathrm{EISR}}(i,j)$≤$r_{\mathrm{EISR}}(i,t)$, then node j is reached by the ripple associated with EISR i. If $e_{\mathrm{EISR}}(i,t)$≥ $\beta_{\mathrm{R}}(j)$, then node j is activated by EISR i, and generates a responding ripple with

$$E_N(j) = \alpha(j) e_{EISR}(i,t), \quad (6)$$

and $e_{\mathrm{N}}(j,t)$= $E_{\mathrm{N}}(j)$.

Step 2.4: If $e_{\mathrm{N}}(i,t\text{-}1)$>0, i=1,…,N_{N}, then update the current radium and point energy of node i in a similar way to EISRs, i.e.,

$$r_N(i,t) = r_N(i,t-1) + s, \quad (7)$$

$$e_N(i,t) = f_{Decay}(E_N(i), r_N(i,t), t). \quad (8)$$

Basically, Eq.(3), Eq.(4), Eq.(7) and Eq.(8) show that, no matter where the ripples originate, they should have the same spreading speed and the same decaying function, just like in the nature world. However, this may be amendable in order to get a more complicated artificial model.

Step 2.5: Check which new nodes are reached by the ripples of other nodes. Suppose $D_{\mathrm{N}}(i,j)$ is the distance between node i and node j. If $E_{\mathrm{N}}(j)$=0 and $D_{\mathrm{N}}(i,j)$≤$r_{\mathrm{N}}(i,t)$, then node j is reached by the ripple generated by node i. If $e_{\mathrm{N}}(i,t)$≥ $\beta_{\mathrm{R}}(j)$, then node j is activated by node i, and generates a responding ripple with

$$E_N(j) = \alpha(j) e_N(i,t), \quad (9)$$

and $e_{\mathrm{N}}(j,t)$= $E_{\mathrm{N}}(j)$. If $e_{\mathrm{N}}(i,t)$≥ $\beta_{\mathrm{L}}(j)$, then a connection between node i and node j is established, i.e.,

$$A(i,j) = A(j,i) = 1 \quad (10)$$

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

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

From the modeling process proposed above, one can see that the network topology is largely determined by the ripple-spreading related parameters. Given different values for the parameters, different topologies will be generated. Therefore it is possible to improve the network topology by optimizing the ripple-spreading related parameters.

Also from the above network modeling process, one can see there are two basic behaviors for each node: being activated and being connected. The evolving of a random network is based on these two behaviors, which are mainly determined by two thresholds: $\beta_{\mathrm{R}}(i)$ and $\beta_{\mathrm{L}}(i)$. These two

behaviors may have 4 combinations, as illustrated in Fig.2, which contribute to the complex evolving behavior of a random network.

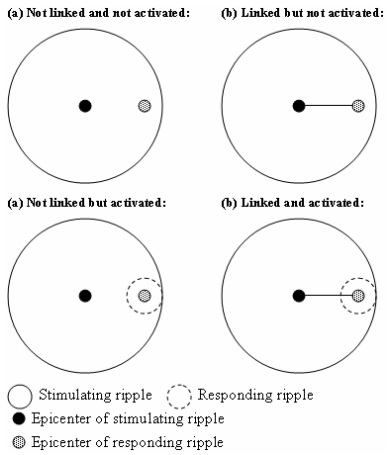


Fig.2. Two behaviors and four combinations.

C. *Three sub-models*

As mentioned in Section I, one motivation of this paper is to develop a deterministic model of random networks, i.e., describe a random network with a few parameters, and once the values of the parameters are given, the network topology will be uniquely determined. This is obviously achieved by the ripple-spreading model proposed above. However, the deterministic model is just one sub-model achievable by the idea of ripple-spreading model. Actually, by slightly modifying some parts in the deterministic model, one can get a semi-deterministic model and a stochastic model with exactly the same idea of ripple-spreading model.

Step 2.3 and Step 2.5 in the network modeling process defines the details of the two behaviors of each node. In the deterministic model, if and only if a threshold is reached, the associated behavior will occur. Therefore, once the distribution of nodes are given and fixed, the network topology is thoroughly determined by the ripple-spreading related parameters, as illustrated in Fig.3(a).

Based on the deterministic model, one can easily introduce some stochastic features as following. In Step 2.3 or Step 2.5, suppose a node is newly reached by a ripple. If the current point energy of the stimulating ripple is above a threshold of this node, then the node behaves as described in Step 2.3 or Step 2.5. In the case where no threshold of the node is reached, no action is defined for the node in the deterministic model, while in the semi-deterministic model, the node may still be activated or connected according to a certain probability function. For instance, when no threshold is reached, i.e.,

$$\beta_R(i) > e_{EISR}(j,t) \text{ or } \beta_R(i) > e_N(j,t), \quad (11)$$

$$\beta_L(i) > e_{EISR}(j,t) \text{ or } \beta_L(i) > e_N(j,t), \quad (12)$$

node i, the node which is newly reached by a ripple, will generate a responding ripple at a probability of $p_R(i)$,

$$p_R(i) = 2^{\omega_R(1-\frac{\beta_R(i)}{e_{EISR}(j,t)})} \text{ or } p_R(i) = 2^{\omega_R(1-\frac{\beta_R(i)}{e_N(j,t)})}, \quad (13)$$

and/or establish a connection at a probability of $p_L(i)$,

$$p_L(i) = 2^{\omega_L(1-\frac{\beta_L(i)}{e_{EISR}(j,t)})} \text{ or } p_L(i) = 2^{\omega_L(1-\frac{\beta_L(i)}{e_N(j,t)})}, \quad (14)$$

where $\omega_R>0$ and $\omega_L>0$ are tail-off coefficients. The probability functions defined by Eq.(13) and Eq.(14) implies that less point energy, smaller probability of being activated or connected, as illustrated in Fig.3(b). Obviously, in a final network topology of the semi-deterministic model, some connections are thoroughly determined by the ripple-spreading related parameters, just like in the deterministic model, while the other connections are largely established in a random manner.

The third sub-model is a stochastic model, where all connections in a final network topology are determined partially by the ripple-spreading related parameters, e.g., the distribution of EISRs, and partially by chance, i.e., according to certain probability functions such as following

$$p_R(i) = 2^{\omega_R(1-\frac{E_{\max}}{e_{EISR}(j,t)})} \text{ or } p_R(i) = 2^{\omega_R(1-\frac{E_{\max}}{e_N(j,t)})}, \quad (15)$$

$$p_L(i) = 2^{\omega_L(1-\frac{E_{\max}}{e_{EISR}(j,t)})} \text{ or } p_L(i) = 2^{\omega_L(1-\frac{E_{\max}}{e_N(j,t)})}, \quad (16)$$

where

$$E_{\max} = \max(\alpha(j)E_{EISR}(i)), \; i=1,\ldots,N_{\text{EISR}}, \; j=1,\ldots,N_{\text{N}}. \quad (17)$$

Clearly, Eq.(17) guarantees no probability is larger than 1.

From the three sub-models discussed above, one can see that, although the ripple-spreading model originally aims to provide a deterministic method to describe random networks, it is completely compatible to the stochastic features of random networks. The stochastic features can be introduced at two levels. One level is composed of the ripple-spreading related parameters. By assigning random values to these parameters, one can get random topologies, even in the deterministic model. The other level includes the probability functions such as given by Eq.(13) to Eq.(16). These probability functions allow stochastic features even for a set of fixed ripple-spreading related parameters, just like in the semi-deterministic model and the stochastic model.

Actually, the ripple-spreading model proposed in this paper has a considerable degree of freedom for further extensions and modifications. One can easily introduce some new parameters into the ripple-spreading process to increase the complexity of the model. For example, to model the preferential attachment behaviour in scale-free networks, one can define a multi-activating behaviour for each node, i.e., each node may be activated more than once, or the point energy of a node can be enforced over time by any ripple whose point energy is above the threshold β_R when reaching the node. The probability functions given in Eq.(13) to Eq.(16) are just some examples, and may be modified according to the requirements on the statistic features of random networks. For instance, The probability functions given in Eq.(13) to Eq.(16) are more likely to generate small-world networks. This is

because the probability decreases as the point energy decays, which is actually the result of the increase in distance. Therefore, basically, nodes closer to each other are more likely to be connected together according to Eq.(13) to Eq.(16). However, for some networks, e.g., hub-and-spoke airline route networks, this feature might be desirable at a local level, but the hub nodes should be reasonably far away from each other. In this case, one may use a semi-deterministic model with the following probability functions

$$p_R(i) = 2^{-\omega_R e_{EISR}(j,t)} \text{ or } p_R(i) = 2^{-\omega_R e_N(j,t)}, \quad (18)$$

$$p_L(i) = 2^{-\omega_L e_{EISR}(j,t)} \text{ or } p_L(i) = 2^{-\omega_L e_N(j,t)}, \quad (19)$$

if no threshold is reached. Applying Eq.(18) and Eq.(19) along with the multi-activating behaviour, hub-connection are more likely to be established between nodes far way from each other. Obviously, a stochastic model with U-shaped probability functions may also exhibit similar statistic features.

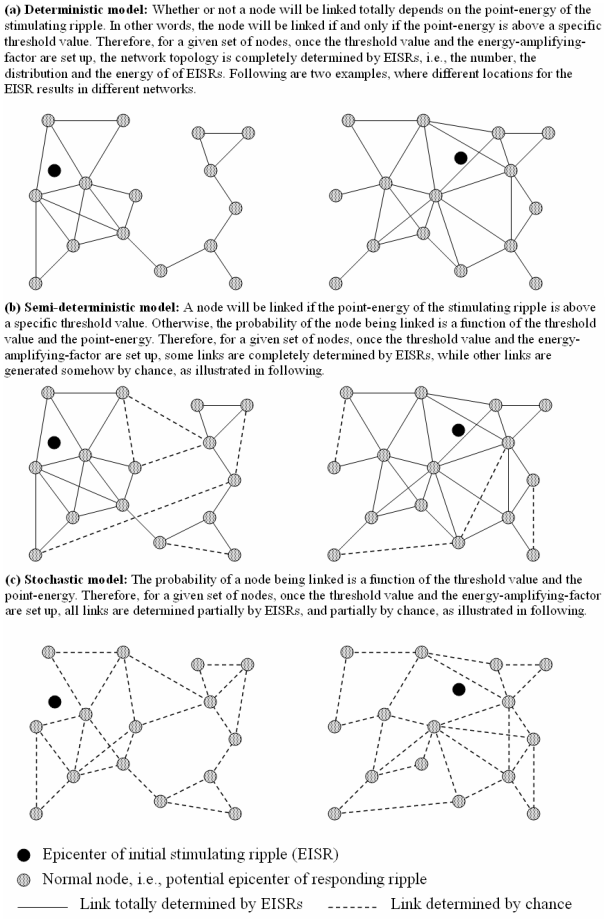


Fig.3. 3 ripple-spreading sub-models of random networks.

III. BINARY GA FOR TOPOLOGY OPTIMIZATION

This section explains, based on the above ripple-spreading model, how to apply the baisc binary GA to the topology optimization of random networks.

A. Basic idea of applying binary GAs to combination problems

As large-scale parallel stochastic search and optimization algorithms, GAs have a good potential to be applied to a wide range of optimization problems [12], [13]. The choice of representation of solutions and the design of evolutionary operators play crucial roles in a successful application of GAs. Permutation representations are usually used when applying GAs to combination problems, but they make it difficult to design effective and efficient evolutionary operators, because these evolutionary operators based on permutation representations are often confronted with feasibility and memory-inefficiency problems.

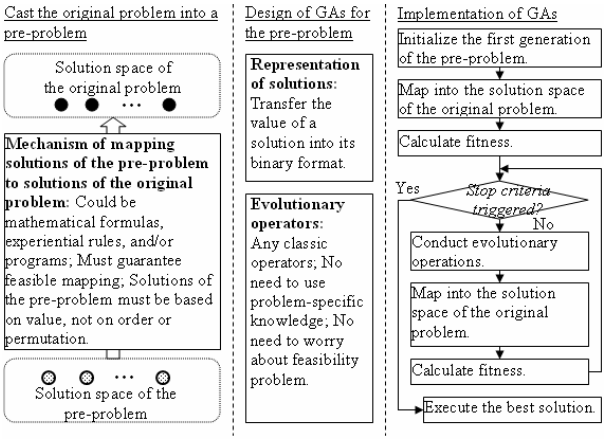


Fig.4. Apply binary GAs to combination problems.

This sub-section explains an idea of using very basic binary GAs to solve combination problems which usually require permutation representations. The binary GAs used must, free of feasibility problem, be compatible to all classic evolutionary operators. To this end, we propose a hybrid GA, the basic idea of which is illustrated in Fig.4. Basic binary GAs are easy to design for those problems where the solutions are based on value, and to such problems all classic evolutionary operators, such as mutation, one-point crossover and uniform crossover, are usually applicable. However, if the solutions are based on combination of elements, such as a random network, it is very difficult to design binary GAs, and crossover is often discarded because it is usually more destructive rather than effective in such problems. In our hybrid GA scheme, an original problem, whose solutions are based on combination and therefore are unsuitable for binary representation, needs to be cast into a pre-problem whose solutions are based on value. Then we design a binary GA for this new pre-problem, and most classic GA techniques can apply straightforwardly. In the implementation of the hybrid GA, the only thing different from a conventional GA is that, before the fitness of a chromosome is calculated, the represented solution to the pre-problem needs to be mapped into the associated solution to the original problem. Obviously the most important and also the most difficult step in the hybrid GA scheme is to design a proper pre-problem, which depends largely on each individual original problem. The ripple-spreading model proposed in Section 2 is a well designed pre-problem for applying binary GAs to optimize the topology of random networks.

B. Chromosome structures

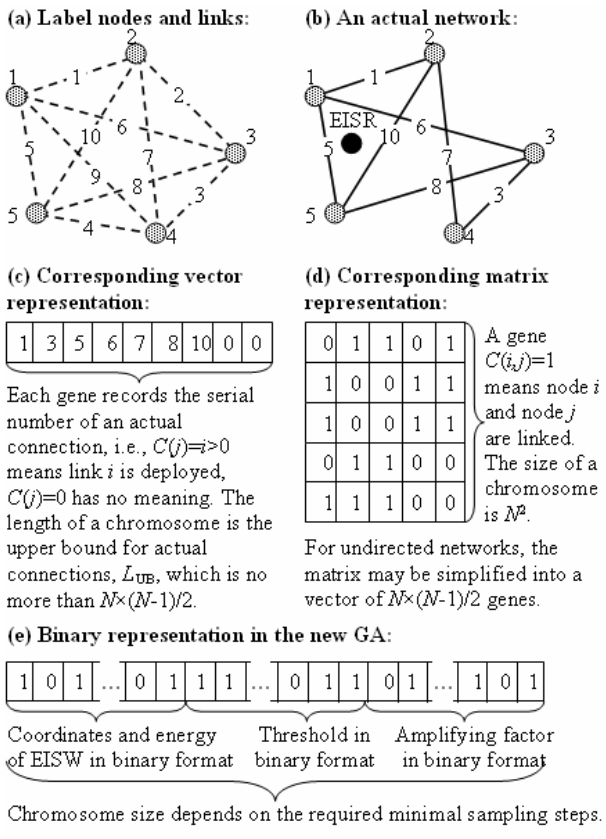


Fig.5. Uniform crossover vs one-point crossover.

In this paper, we only consider deterministic chromosome structures, i.e., a chromosome must be able to determine a unique network topology. For comparative purposes, we give two widely used structures based on actual connections between nodes. One is a vector representation, where each gene records the serial number of an actual connection, as illustrated in Fig.5(c); the other is a matrix representation, where a gene $C(i,j)=1$ indicates node i and node j are connected, and $C(i,j)=0$ otherwise, as illustrated in Fig.5(d). Since both the vector structure and the matrix structure are directly based on the link combination information, they both belong to permutation representation. In our GA, a chromosome has no direct link to the represented network, because it is simply a binary string of the values for those ripple-spreading related parameters, as shown in Fig.5(e). Table I summarizes the features of different structures, from which one can see that: (I) the vector representation has no serious memory-inefficiency problems as long as the upper bound for actual connections $L_{UB}<<N\times(N-1)/2$, but it may be confronted with feasibility problems during evolutionary operations, particularly, the traditional crossover, such as one-point crossover and uniform crossover, can easily violate the feasibility constraint, unless some additional computationally expensive feasibility checking and repairing operations are introduced; (II) basically, the matrix representation has no serious feasibility problems, but it may suffer from a huge memory demand when a large scale network is under consideration, for instance, if N=1000, then the size of single chromosome is 1Mb, and a generation with a population of 1000 (a larger N requires a bigger population in order to maintain the level of solution quality) demands a memory capacity of at least 1Gb, which is not affordable for many standard personal computers, let along there are many real networks having millions of nodes; (III) owing to the ripple-spreading model, a chromosome in our new GA is simply a binary string of encoded parameter values, and has nothing explicit to do with the represented network topology, therefore it has no feasibility constraints, and the chromosome size is independent of the network scale and then always manageable, for instance, for each ripple-spreading related parameter, a 64-bit binary string can provide a very satisfactory sampling quality, and therefore the memory demand for a chromosome is just $64\times N_P$, given there are N_P ripple-spreading related parameters used in the model (obviously $N_P<<N$, particularly in the case of complex networks). As a conclusion, compared with the permutation representations, the binary chromosome in our new GA is free of both feasibility and memory-inefficiency problems. Further more, as will be discussed later, this binary representation of parameter values is compatible and friendly to all classic evolutionary operators and techniques.

TABLE I
FEATURES OF DIFFERENT REPRESENTATIONS

	Meaning of a gene	Meaning of a chromo.	Size of a chromo.	Constraints
Vector represen-tation	$C(j)=i>0$ means link i is deployed	Which links are deployed	$L_{UB}\le N\times(N-1)/2$	$C(j)\neq C(h)$ if $h>j$, $C(h)>0$
Matrix representtation	$C(i,j)=1$ means node i and node j are linked	Which links are deployed or not	N^2	$C(i,j)=C(j,i)$ undirected networks
Binary representtation	A digit bit in the binary string	Ripple-spread-ing related parameters	Relying on sampling steps	None

C. Evolutionary operators

Thanks to the ripple-spreading model given in Section 2, the original network topology optimization problem, which requires permutation representation, is transformed into a pre-problem whose solution is simply composed of the values of those ripple-spreading related parameters. As a result, no effort is needed to design any problem-specific evolutionary operator, but all classic techniques can apply straightforwardly. The mutation operator simply reverses the value of a randomly chosen gene, i.e.,

$$C(i)=1-C(i). \quad (20)$$

Crossover may cause feasibility problems to permutation representations, and therefore is often discarded in combination problems. Fortunately, in the binary GA, we can use any traditional crossover operators, such as one-point crossover and uniform crossover. One-point crossover uses two parents to produce two offspring: randomly choose a location where parent chromosomes shall be divided, and then exchange the first section of one parent with the first section of the other parent. Uniform crossover uses two

parents to produce only one offspring, and the principle is simple: the *i*th gene in the offspring inherits the *i*th gene of either parent at a half-and-half chance [14]. As illustrated in Fig.6, with the binary representation, both crossover operators can automatically keep sections that are same in parent chromosomes. On the other hand, uniform crossover offers more freedom to evolve chromosomes than one-point crossover does and in this way maximizes the sampling of non-shared genes between parents.

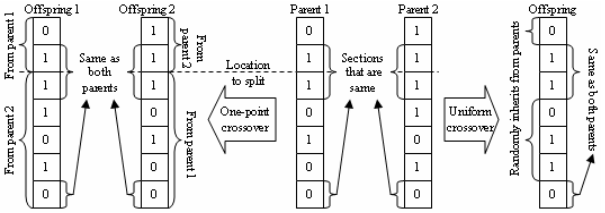


Fig.6. Uniform crossover vs one-point crossover.

D. Heuristic rules

In permutation representation based GAs for combination problems such as network topology optimization, many problem-specific heuristic rules are integrated into their evolutionary operations. Differently, in the evolutionary operations of our binary GA with the ripple-spreading model, we mainly focus on some pure GA-related heuristic rules. For example, evenly distribute a certain proportion of chromosomes within the solution space to the pre-problem when initializing a generation; online adjust mutation probability and crossover probability according to the fitness of each individual chromosome as well as the overall fitness level of the current generation of chromosomes [15]; dynamically restrict the mutation operation to a certain part of a chromosome based on its fitness [16]; some GA-related parameters may be included in the chromosome structure and then get evolved along with those ripple-spreading related parameters, just like genetic strategies do [13]. As for problem-specific knowledge, it should be mainly integrated into the ripple-spreading model.

IV. Preliminary Simulation Results

This section only provides some preliminary simulation results in order to demonstrate the potential of the proposed ripple-spreading random network model and the associated binary GA for the optimization of network topology. For the sake of simplicity, in the simulation, the deterministic sub-model is used, and it is assumed N_{EISR}=4, $T_{EISR}(i)$=0, $E_{EISR}(i)$=E_{EISR}, i.e., all EISRs have the same active time and the same initial energy, and N_N=100, $\alpha(i)$=α, $\beta_R(i)$=$\beta_L(i)$=β, i.e., all nodes have the same amplifying factor, and the same threshold both for being activated and for being linked. In the simulation, we only evolve the distribution of EISRs, i.e., $(x_{EISR}(i),y_{EISR}(i))$, in order to generate evenly distributed connections between nodes. Assuming the desirable average degree is d_{ave}=6, then the mathematical description of the optimization problem in the simulation is

$$\min_{(x_{EISR}(i),y_{EISR}(i)),i=1,\ldots,4} J = \min_{(x_{EISR}(i),y_{EISR}(i)),i=1,\ldots,4} \sum_{j=1}^{N_N}(d(j)-d_{ave})^2 \quad (21)$$

where $d(j)$ is the degree of node j. Fig.7 gives some networks resulted from different distributions of EISRs, where one can see that the distribution of EISRs can effectively affect the network topology, and therefore may be used as tuning parameters to be evolved by GAs in order to improve the topology in terms of concerned network properties. From Fig.7, one may notice that the ripple-spreading model of random networks is particularly suitable for describing spatial embedded networks, which have a solid real-world background. For instance, communications networks may involve range-dependent links; social networks may involve distance-limited interactions between agents existing in some (possibly abstract) space; transport networks have an obvious spatial embedding; certain spatial patterns often exist in breakout of infectious diseases while, more generally, technological/commercial networks frequently feature some cost-per-distance constraint on connectivity.

However, this paper only reports the very initial work on the ripple-spreading model of random networks and the design of the associated binary GAs. Clearly, in the optimization problem defined by Eq.(21), we do no use the full range of ripple-spreading related parameters as input, neither do we employ an objective function especially constructed for certain real-world networks. Therefore, the true potential and usefulness of our ripple-spreading model and GA to optimize real complex networks are not fully understood or exploited. As will be discussed in the following section, more efforts, improvements, statistical analyses and comparative experiments need to be carried out in order to complete the picture of the proposed methodology for studying random complex networks.

V. Conclusions and Future Work

This paper attempts to develop a methodology to apply basic binary Genetic Algorithms (GAs) for the general topology optimization of random complex networks. When GAs are used to optimize random networks, permutation representation, memory inefficiency, stochastic modeling, and difficulties in the design of problem-specific evolutionary operators are some common problems. Inspired by the natural ripple-spreading phenomenon, this paper reports a deterministic method to model random networks. Some ripple-spreading related parameters can completely determine the topology of a random network. In other words, the randomness of complex networks is transferred to the randomness of these ripple-spreading related parameters. As a result, the very basic binary GA, free of memory-inefficiency problems, compatible to all classic evolutionary operators, can be applied straightforwardly to optimize these parameters in order to improve the topology of random networks. The potential of the proposed model and GA is illustrated by preliminary simulation results. Future research include: (I) Study the ripple-spreading model in

more depth, e.g., conduct a full-scale statistic analysis of the model in terms of complex network properties; (II) Develop some problem-specific sub-models, e.g., to generate small-world networks and to generate scale-free networks probably require different sets of ripple-spreading related parameters; (III) Test on some real-world networks and compare with other models and methods; (IV) Extend the basic idea, i.e., develop a ripple-spreading model, and then use the basic binary GA to optimize, to other combination problems such as scheduling problems.

REFERENCES

[1] Boccaletti, S., V. Latora, M.Y. Chaves, and M. Hwang, D.U., "Complex networks: Structure and dynamics". *Physics Reports* vol. 424, 175-308, 2006.

[2] Albert, R. and A.L. Barabasi, "Statistical mechanics of complex networks". *Reviews of Modern Physics*, vol. 74, 47-97, 2002.

[3] Bollobas, B., *Random Graphs*, Academic, 1985.

[4] Erdos, P., and A. Renyi, *Publ. Math.* (Debrecen), vol. 6, 290, 1959.

[5] Aiello, W., F. Chung, and L. Lu, *Proceedings of the 32nd ACM Symposium on the Theory of Computing*, ACM, New York, 2000.

[6] Newman, M.E.J., *Phys. Rev.*, vol. 64, 025-102, 2001.

[7] Barnett, L., E. Di Paolo and S. Bullock, "Spatially Embedded Random Networks", *Phys. Rev.*, in press, 2008.

[8] Xin, Z.H. and H.J. Zhang, "Neural Network and Genetic Algorithms for Topology Optimization of the CCS7 Network". *International Transactions in Operational Research*, 9, 427-436, 2002

[9] El-Alfy, E.-S.M., "Network Topology Design Using Genetic Algorithms". *Proceedings of IEEE International Conference on Computer Systems and Applications*, 1059 –1065, Washington, DC, USA, 2006.

[10] Hu, X.B. and E. Di Paolo, "An Genetic Algorithm based on Complex Networks Theory for the Management of Airline Route Networks", *International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, Acireale, Italy, 08-10, Nov 2007.

[11] Isaacs, A., T. Ray, and W. Smith, "Novel evolutionary algorithm with set representation scheme for truss design". *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore, 25-28, Sep 2007.

[12] Holland, J.H. *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.

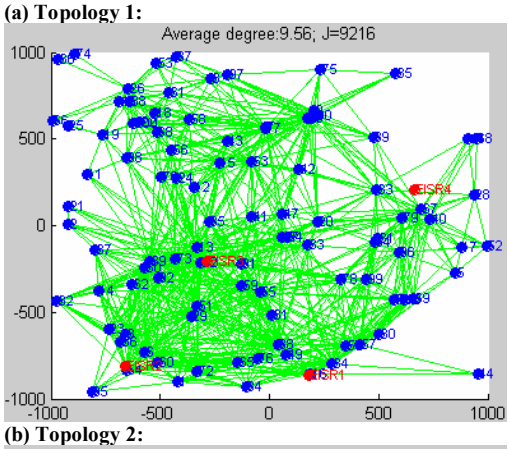
[13] Eiben, A.E. and J.E. Smith, *Introduction to Evolutionary Computing*, Berlin, Germany: Springer-Verlag, 2003.

[14] Sywerda, G., "Uniform crossover in genetic algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms*, USA, 1989.

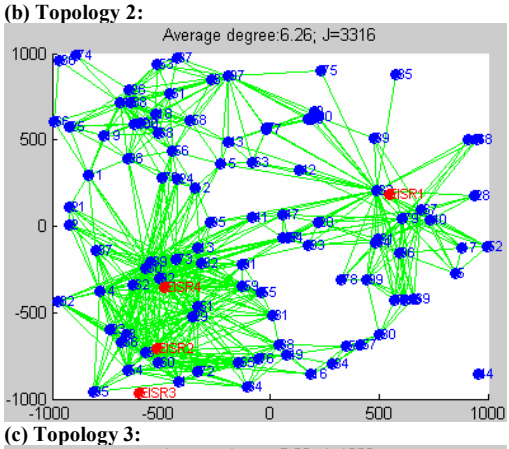
[15] Zhang, J., H.S.H. Chung and W.L. Lo, "Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms", *IEEE Transaction on Evolutionary Computation*, vol.11, pp. 326-335, 2007.

[16] Hu, X.B. and S.F. Wu, "Self-Adaptive Genetic Algorithm Based on Fuzzy Mechanism", *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore, 25-28, Sep 2007.

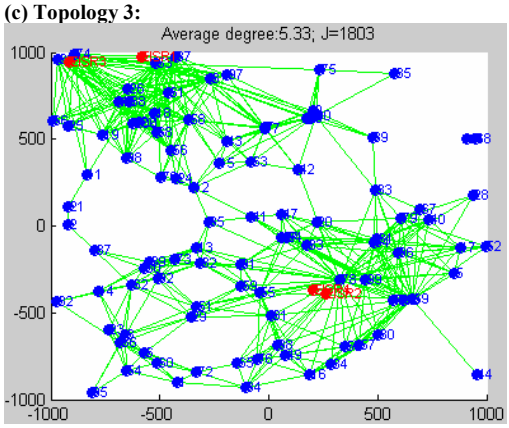
(a) Topology 1:



(b) Topology 2:



(c) Topology 3:



(d) Topology 4:

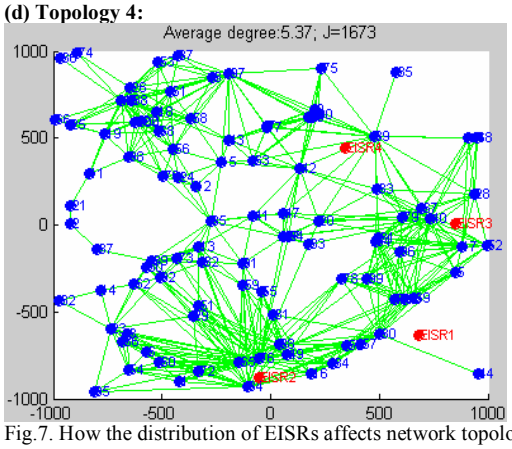


Fig.7. How the distribution of EISRs affects network topology.