

Dual-stage constructed random graph algorithm to generate random graphs featuring the same topological characteristics with power grids



Shiqian MA^{1,2}, Yixin YU², Lei ZHAO²

Abstract It is a common practice to simulate some historical or test systems to validate the efficiency of new methods or concepts. However, there are only a small number of existing power system test cases, and validation and evaluation results, obtained using such a limited number of test cases, may not be deemed sufficient or convincing. In order to provide more available test cases, a new random graph generation algorithm, named “dual-stage constructed random graph” algorithm, is proposed to effectively model the power grid topology. The algorithm generates a spanning tree to guarantee the connectivity of random graphs and is capable of controlling the number of lines precisely. No matter how much the average degree is, whether sparse or not, random graphs can be quickly formed to satisfy the requirements. An approach is developed to generate random graphs with prescribed numbers of connected components, in order to simulate the power grid topology under fault conditions. Our experimental study on several realistic power grid topologies proves that the proposed algorithm can quickly generate a large

number of random graphs with the topology characteristics of real-world power grid.

Keywords Power grid topology, Dual-stage constructed random graph (DSCRG) algorithm, Random graph generation, Connectivity, Average degree, Connected component

1 Introduction

Simulations based on standard models are often used as part of the engineering design process to examine or test the innovative concepts, novel algorithms or control schemes proposed by researchers. In the electric power industry, studies based on the IEEE benchmark systems [1] defy enumeration. For example, the effectiveness of the proposed algorithm for intentional islanding was tested on the PG&E 69-node system in our previous work [2]. It would be more sufficient and convincing to evaluate the performance of the proposed algorithm based on a large number of test cases. However, the number of standard models is very small (less than one hundred), even including other open source real-world power system models [3]. There is a lack of power grid models to simulate a range of phenomena in power systems [4–7]. In order to provide a statistically significant number of examples to explore the variability of phenomena under study, a fundamental requirement is to generate a large number of test power systems featuring the same topology characteristics as realistic or standard test systems.

Connectivity is the most important property of power system topologies because the primary function is to supply energy to customers. A power system is not always in a normal operation state. When faults occur, it is possible

CrossCheck date: 3 May 2017

Received: 3 May 2016 / Accepted: 3 May 2017 / Published online: 2 September 2017

© The Author(s) 2017. This article is an open access publication

✉ Yixin YU
yixinyu@tju.edu.cn

Shiqian MA
msq112358@126.com

Lei ZHAO
1401281038@qq.com

¹ State Grid Tianjin Electric Power Research Institute, Tianjin 300384, China

² Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China



that the transmission systems are disconnected and the distribution systems are broken into several regions by the fault elements. From the point of view of graph theory, the topology of a power system under a fault condition is a graph with multiple connected components. The reconfiguration of distribution feeders will create a large number of combinations of switch states in a post-fault system, which means a great variety of distribution network structures and corresponding graphs. However, there is no appropriate random graph which is capable of fully exhibiting such characteristics of distribution networks.

Sparsity is another main feature of power grid topologies. The average degree (i.e. the average value of the number of lines connected to each node) of a realistic power grid ranges from 2 to 5, normally between 2 to 3, and does not scale with the grid size [8]. For example, the average degrees of the power grids in northern China and western US are 2.23 [9] and 2.67 [10], respectively, while the degree of medium/low voltage grids in the Netherlands is 2.18 [11]. Generally, it is difficult to generate a connected random graph with a very low average degree, because a sparser graph has a lower possibility of being connected. Yet the distribution of node degrees [12] of generated random graphs must fit with real-world power grid topologies, and this must be achieved without consuming excessive time in the process of generating random graphs.

It is necessary to research how to generate in a reasonable time frame a large number of sparse random graphs with prescribed numbers of connected components, and with the appropriate distribution of node degree, to substitute for real-world power grid topologies.

A series of synthetic power system models were designed to assist the study of power system blackouts by Carreras *et al.* [4–6]. The purpose was to avoid incomplete and inaccurate historical blackout data affecting the credibility of the analysis results. Synthetic power system models with tree structure, quadrilateral structure and hexagonal structure were implemented to simulate large power outages. Parashar *et al.* [7] designed a ring model of power grid, when testing the proposed continuum model of electromechanical dynamics in large-scale power systems. They attempted to use a simple ring structure to reflect the essential characteristics of electromechanical wave propagation in the power system. However, real-world power systems are not uniform, and may contain a large number of irregular structures, so the ideal models used in [4–7], including tree, ring or other regular networks, are all incapable of fully reflecting the topological properties of realistic power grids. The results obtained using such regular network models may be biased or incomplete. Therefore, we need to establish a more rigorous model of power grid topology.

The Erdős-Rényi (ER) random graph [13] is often used as a test case to substitute for a real network topology. In the ER model, the probability of any line connecting two arbitrary nodes is p . Thus, in a random graph with n nodes, the expected number of lines is $p[n(n-1)/2]$. The expected average degree in an ER random graph is $p(n-1)$. We can adjust the input value p to obtain the average degree that we need. In [14, 15], an improved model called ZER was developed based on the ER model. It can shorten the computation time by skipping some specific lines. Although the model is more advanced than the ER model, both of them have the disadvantage that they give no assurance about the connectivity of the generated random graphs. When $n \rightarrow \infty$, if p is larger than a critical value p_c , ($p_c \propto \ln n/n$), then the generated random graphs are connected in most cases. Conversely, when p is small, the graphs are hardly connected at all. As a result, the generation processes needs to be repeated many times until a connected random graph is generated, which is inefficient.

The work by Watts and Strogatz [10] first statistically modeled the power grid as a small-world graph, which is generated by starting with a regular ring lattice, then rewiring some local lines with a probability p to arbitrary nodes chosen uniformly at random from the entire graph. A small-world model of a power transmission system was established in [16] to evaluate a disturbance in it. Although the small-world characteristic of the power grid is highlighted, the average degree of the Watts-Strogatz small-world model can only be an even number, which is not in accord with the node degree heterogeneity of the real-world power grid.

Wang *et al.* [17] proposed a random topology generation algorithm for studying the scale of communication needs and the performance of the combined electric power control and communication networks. The algorithm consists of three steps: (a) randomly place n nodes in a fixed area, using a normal or Poisson distribution; (b) given the distance constraint d , connect each node to all its neighboring nodes within the distance d ; (c) check whether the generated topology is connected, if not, repeat the step (a) and (b). The average degree of the random topology generated by this algorithm is related to the distance d and the distribution of the nodes, and is not controllable. Based on the above model, an improved hierarchical random topology model was proposed in [18], as an analytical tool to examine the efficiency of any networked control architecture in smart grid applications. The generation algorithm consists of two steps: (a) generate a number of sub-networks, where in each sub-network every node is connected with k nodes within the distance d (where k is a positive integer greater than one); then determine which lines need to be rewired by adjusting three parameters α , β , and p_{rw} ;

(b) connect all the sub-networks generated in step (a). The average degree of the above random topology is k , which can only be an integer. The topological characteristics depend on the four parameters d , α , β , and p_{rw} .

The above works demonstrate the importance of generating random topologies for power system studies. However, the existing random topology models of power grids have several shortcomings: (a) the generated topologies are not necessarily connected; (b) the average degree of the generated topologies cannot be controlled precisely; (c) the generated topologies differ from real-world power grid topologies in the node degree distribution; (d) the number of connected components contained in the generated topology is equal to one. Therefore, in this paper we focus on the above problems, and propose a dual-stage constructed random graph (DSCRG) generation algorithm to model power grid topologies, which is superior to the existing algorithms. The proposed algorithm can guarantee the connectivity of the generated random graph through forming a spanning tree; the number of lines in the generated random graph is completely determined by the required average degree; the node degree distribution fits well with real-world power grid topologies; and the number of connected components in the generated random graph is controllable. The effectiveness of the proposed algorithm is validated by experimental studies.

The remainder of this paper is structured as follows. The next section introduces the definitions and notation used in this paper. Section 3 describes the DSCRG algorithm. We conduct an extensive empirical comparison of the performance of proposed algorithm and several existing algorithms in Section 4. We conclude in Section 5.

2 Definitions and notation

- 1) Graph: A graph G is a representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by mathematical abstractions called nodes (also called vertices or points), and the links that connect some pairs of nodes are called lines (also called arcs or edges).
- 2) Undirected graph: An undirected graph $G(V, E)$ is a graph in which lines have no orientation; V is the set of nodes in G ; E is the set of lines.
- 3) Connected graph: In an undirected graph, an unordered pair of nodes $\{x, y\}$ is called connected if a path leads from x to y . Otherwise the unordered pair is called disconnected. A connected graph is an undirected graph in which every unordered pair of nodes in the graph is connected. Otherwise, it is called a disconnected graph.

- 4) Connected component: A connected component is a connected sub-graph of graph G . A connected graph has only one connected component, that is, itself, while an unconnected graph has multiple connected components.
- 5) Node degree: The degree of a node is the total number of lines connected to the node.
- 6) Average degree: The average degree (denoted by $\langle k \rangle$) of a graph G is another measure of how many lines are in set E (denoted by e) compared to the number of nodes (denoted by n) in set V . Because each line is incident to two nodes and counts in the degree of both nodes, the average degree and number of lines of a graph is:

$$\langle k \rangle = 2e/n \Leftrightarrow e = n\langle k \rangle/2 \quad (1)$$

- 7) Node degree distribution: The node degree distribution (degree distribution for short) of a graph can be described by a distribution function $P(k)$, which represents the probability that the degree of an arbitrary node is exactly k . A completely random graph has a distinctive characteristic that the degree distribution is similar to a Poisson distribution with a mean value $\langle k \rangle$ [19].
- 8) Kullback–Leibler divergence: In probability theory and information theory, the Kullback–Leibler divergence [20] (also information divergence, information gain, relative entropy, KLIC, or KL divergence) is a measure of the difference between two probability distributions P and Q . It is not symmetric in P and Q . In applications, P typically represents the “true” distribution of data, observations, or a precisely calculated theoretical distribution, while Q typically represents a theory, model, description, or approximation of P . Specifically, the Kullback–Leibler divergence of Q from P , denoted $D_{KL}(P||Q)$, is a measure of the information gained when one revises one’s beliefs from the prior probability distribution Q to the posterior probability distribution P . For discrete probability distributions P and Q , the Kullback–Leibler divergence of Q from P is defined to be:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \quad (2)$$

- 9) Complete graph: A complete graph is a graph in which each pair of graph nodes is connected by a line. The complete graph with n graph nodes is denoted by K_n . The average degree of K_n denoted by $\langle k_{K_n} \rangle$ is:

$$\langle k_{K_n} \rangle = 2 \cdot \frac{n(n-1)}{2} / n = n - 1 \quad (3)$$



- 10) Forest: A forest (denoted by F) is an acyclic graph (i.e., a graph without any graph cycles). Forests therefore consist only of (possibly disconnected) trees, hence the name “forest”. The line number of a forest with n nodes and c connected components is $n - c$. The average degree of a forest denoted by $\langle k_F \rangle$ is:

$$\langle k_F \rangle = 2(n - c)/n \quad (4)$$

- 11) Tree: A tree (denoted by T) is a connected acyclic graph. Thus, each component of a forest is a tree, and any tree is a connected forest. The line number of a forest with n nodes is $n - 1$. The average degree of a tree denoted by $\langle k_T \rangle$ is:

$$\langle k_T \rangle = 2(n - 1)/n \quad (5)$$

- 12) Spanning forest: A spanning forest is a forest that contains every node of G such that two nodes are in the same tree of the forest when there is a path in G between these two nodes.
- 13) Spanning tree: A spanning tree of a connected graph G is a maximal set of lines of G that contains no circle or a minimal set of lines that connect all nodes.
- 14) Node-branch incidence matrix: If a power network has n nodes and m branches, its node-branch incidence matrix $A(n \times m)$, can be formulated as follows:

$$A = \begin{cases} A_{i,b} = 1; A_{j,b} = -1 & \text{if } b^{\text{th}} \text{ branch is between} \\ & \text{node } i \text{ and } j \\ A_{k,b} = 0 & \text{if } k \neq i, j \end{cases} \quad (6)$$

- 15) Node admittance matrix: Also network admittance matrix, is defined as follows:

$$Y = AY_b A^T \quad (7)$$

where Y_b represents the branch admittance matrix, and can be derived from branch impedance matrix; A represents the above mentioned node-branch incidence matrix.

3 DSCRG algorithm

In this section, we firstly propose an algorithm to generate a connected random graph (that contains only one connected component), then generalize the algorithm to generate random graphs with prescribed numbers (one or multiple) of connected components. We call the algorithm “dual-stage constructed random graph” (DSCRG for short).

As an artificial network, the main task of a power system is to deliver electrical energy to customers, which means

connecting the power sources and loads through power lines. Hence the connectivity is a key feature of power system topologies. Reviewing the history of power systems, we find the power system topologies in the past can only meet the requirement of connectivity. From the point of view of graph theory, the earliest power system topology is a tree-like structure. We can easily define the power source and the loads as the root node and the leaf nodes, respectively. There is only one path between the root node and each leaf node. As power system technology was developed, and customers required more reliable electricity, the topology of power systems began to change. Transmission systems are designed as reticular structures. A number of tie lines or tie switches are installed in distribution systems, which are implemented as backup lines or switches to transfer loads when faults occur. Through the above analysis, we design an algorithm to obtain connected random topologies for simulating real-world power grids. Firstly, we stochastically generate a spanning tree that contains all the nodes in the power grid, and then add a certain number of lines to meet the requirement of the average degree of the power grid topology.

However, the power system is not always in its normal operation state. When faults occur, it is possible for transmission systems to be disconnected and the distribution systems will be broken into several regions by the fault elements. From the point of view of graph theory, the topology of power system in a fault condition is a graph with multiple connected components. Fault conditions of power systems need more attention by system operators and researchers, but random topologies with only one connected component are not adequate for simulating a fault condition. We need to develop an approach for generating random graphs with prescribed numbers (one or multiple) of connected components.

3.1 Generation of random graph with one connected component

3.1.1 Computations

Consider a connected random graph G , for which the node number is n and the average degree is $\langle k \rangle$, where $n > 1$. The value of $\langle k \rangle$ is between the average degree of a spanning tree with n nodes $\langle k_T \rangle$ (derived from (5)) and the average degree of a complete graph with n nodes $\langle k_{Kn} \rangle$ (derived from (3)), which means

$$2(n - 1)/n \leq \langle k \rangle \leq n - 1 \quad (8)$$

Thus, a connected graph with n nodes can be regarded as a spanning tree plus a certain number of expanded lines. This number, denoted by S , equals the difference between e and the line number of the spanning tree. By (1), e can be

calculated, and the line number of the spanning tree is $n - 1$. Thus, the number of lines to be expanded S is

$$S = \langle k \rangle \cdot n/2 - (n - 1) \tag{9}$$

In this way, the process of generating a connected random graph can be divided into two stages: randomly generating a spanning tree with n nodes and expanding the spanning tree by randomly adding a certain number of lines. In the second stage, the number of optional lines denoted by R is equal to the line number of K_n minus the line number of the spanning tree, which is

$$R = n(n - 1)/2 - (n - 1) \tag{10}$$

Thus, in the second stage, we need to select S lines randomly from R lines.

3.1.2 Procedure

We propose an algorithm to generate a random graph with one connected component. The procedure is explained below.

3.1.2.1 First stage: Randomly generating a spanning tree with n nodes

- 1) A random graph is represented by a node adjacency matrix $A(n \times n)$. The elements of matrix A are 0 or 1, where “0” means that no line exists between two nodes while “1” is the contrary. Assign A to a zero matrix at the beginning.
- 2) The serial numbers of n nodes are $\{1, 2, \dots, n\}$. Pick one node from n nodes at random, and put it into set X . X contains the nodes belong to the spanning tree.
- 3) Put the remaining $(n - 1)$ nodes into set Y . Y contains the nodes don’t belong to the spanning tree.
- 4) Determine whether Y is empty, and if so, then go to stage 2 and do step 1, or otherwise proceed to step 5.
- 5) Randomly select one node from each of set X and Y , denoted by x and y , respectively.
- 6) Connect x and y with a line, i.e., set $A(x, y) = 1$ and $A(y, x) = 1$.
- 7) Remove node y from set Y , and put it into set X , then return to step 4.

3.1.2.2 Second stage: Randomly adding the correct number of lines to the spanning tree

- 1) Compute S by (9).
- 2) Compute R by (10).
- 3) Utilize i and j to represent two different nodes, and let $i = 1, j = 2$.

- 4) Determine whether there is a line between i and j , i.e., $A(i, j) = 0$. If so, proceed to step 5, otherwise proceed to step 10.
- 5) Generate a uniform random number θ that is between 0 and 1.
- 6) Determine whether θ is less than the ratio of S and R , and if so, then proceed to step 7, otherwise proceed to step 8.
- 7) Connect i and j with a line, i.e. set $A(i, j) = 1$ and $A(j, i) = 1$. Let $S = S - 1$.
- 8) Let $R = R - 1$.
- 9) Determine whether S is zero. If so, then exit the loop, and matrix A is the resulting generated random graph, otherwise proceed to step 10.
- 10) Let $i = i + 1, j = j + 1$.
- 11) Determine whether i equals $n - 1$. If so, then exit the loop, and matrix A is the resulting generated random graph, otherwise proceed to step 5.

3.2 Generation of random graph with a prescribed number of connected components

Based on the algorithm presented in section 3.1, we propose a more generalized algorithm to generate the random graph with a prescribed number (one or multiple) of connected components. To obtain a random graph with n nodes and c connected components, we intend to generate the connected components one by one, and then form them into a whole graph. The first thing we should do is to determine the value of node number and average degree for each connected components.

3.2.1 Computations

- 1) The value of node number for each connected component

The sum of the node numbers of all the connected components is n . Let n_i be the node number of connected component i , then we have (11):

$$\begin{cases} n = \sum_{i=1}^c n_i & i = 1, 2, \dots, c \\ n_i \geq 2 \end{cases} \tag{11}$$

We randomly generate n_i for all the connected components with the constraint of (11).

- 2) The value of average degree for each connected component



We attempt to make the average degree of each connected component be the same as that of the whole graph, but the node number of each connected component is generated randomly, which may cause some component to have a small n_i , so that it is impossible for its average degree to reach the value of $\langle k \rangle$, i.e.

$$\begin{cases} \exists i \in \{1, 2, \dots, c\} \\ \text{s.t. } n_i - 1 < \langle k \rangle \end{cases} \quad (12)$$

Meanwhile some other components may have larger n_j . This means we have to increase the number of lines in the larger components to make up the shortage of lines in the smaller components, so that the average degree of the entire graph can reach the given value of $\langle k \rangle$. If the given $\langle k \rangle$ is out of an appropriate range, then the average degree of the generated random graph may not be the required value. For example, given that $n = 6, c = 2$, and $\langle k \rangle = 7/3$, then there needs to be 7 lines in the random graph. First, we randomly generate the node number of each connected component, and assume that the result is $n_1 = 3$, and $n_2 = 3$. At this moment, there can be 6 lines at most in the two components, thus the average degree can be 2 at most, which doesn't reach the given value $7/3$. In order to avoid that situation, we discuss the appropriate range of the given value $\langle k \rangle$. Hence, we present a Proposition to restrict the range of $\langle k \rangle$.

Proposition *The average degree of a random graph with n nodes and c connected components has to satisfy the following inequality:*

a) Prove the right side of (13)

$$2(n - c)/n \leq \langle k \rangle \leq n/c - 1 \quad (13)$$

When every connected component is a complete sub-graph, the line number of the entire graph is maximized, denoted by e_{Kn} :

$$e_{Kn} = \sum_{i=1}^c n_i(n_i - 1)/2 \quad (14)$$

Due to (11), e_{Kn} is minimized when the node number of each connected component is equal, i.e.

$$n_i = n/c, \quad i = 1, 2, \dots, c \quad (15)$$

Substitute (15) into (14), and we obtain (16) as follows:

$$\min(e_{Kn}) = c \cdot \frac{n}{c} \left(\frac{n}{c} - 1 \right) / 2 = n(n - c) / 2c \quad (16)$$

According to (8), the upper bound of $\langle k \rangle$ is the minimum value of $\langle k_{Kn} \rangle$, which is obtained by substituting (16) into (1). So the upper bound of $\langle k \rangle$ is:

$$\langle k \rangle \leq \min(\langle k_{Kn} \rangle) = \frac{2\min(e_{Kn})}{n} = \frac{2n(n - c)}{2cn} = \frac{n}{c} - 1 \quad (17)$$

This proves the right side of (13).

b) Prove the left side of (13)

The spanning forest of a random graph has the minimum number of lines. In other words, when all of the connected components are trees, the line number of the random graph is minimized. The line number is denoted by e_f where

$$e_f = \sum_{i=1}^c (n_i - 1) \quad (18)$$

Substitute (11) into (18), and then we obtain (19):

$$e_f = n - c \quad (19)$$

According to (8), the lower bound of $\langle k \rangle$ is $\langle k_f \rangle$, which is obtained by substituting (19) into (1). So the lower bound of $\langle k \rangle$ is

$$\langle k \rangle \geq \langle k_f \rangle = 2e_f/n = 2(n - c)/n \quad (20)$$

This proves the left side of (13), and the Proposition is proved through (a) and (b).

Inequality (13) shows the value range of average degree of a random graph with n nodes and c connected components. Based on (13), we calculate the average degrees of all the components as follows:

i) $\forall i \in \{1, 2, \dots, c\}$, s.t. $n_i \geq \langle k \rangle + 1$, let
 $\langle k_i \rangle = \langle k \rangle \quad (21)$

ii) $\forall j \in \{1, 2, \dots, c\}$, s.t. $n_j \geq \langle k \rangle + 1$, let the j^{th} connected component be a complete sub-graph. Then the average degree of j^{th} connected component is
 $\langle k_j \rangle = n_j - 1 \quad (22)$

iii) The number of lines of all of the remaining connected components except for the j^{th} component is denoted by e_r , and equals the line number of the entire random graph minus the line number of the j^{th} connected component, i.e.
 $e_r = \langle k \rangle \cdot n/2 - n_j(n_j - 1)/2 \quad (23)$

iv) The average degree of all of the remaining components except for the j^{th} component is denoted by $\langle k_R \rangle$,
 $\langle k_R \rangle = \frac{2e_r}{n - n_j} = \frac{\langle k \rangle \cdot n - n_j^2 + n_j}{n - n_j} \quad (24)$

v) Continue to determine whether the node numbers of the remaining components except for component j are less than $\langle k_R \rangle + 1$, until the average degrees of all of the components are obtained.

3.2.2 Procedure

Consider that the number of connected components is c (c is a positive integer); the node number is n ($n \geq 2c$); and the average degree is $\langle k \rangle$, whose value range is restricted by (13). The procedure of the generalized algorithm is:

- 1) Randomly generate the node numbers of all of the connected components, and sort them into ascending order, denoted by n_1, n_2, \dots, n_c .
- 2) Let $j^{(0)} = 0, l = 1, \langle k^{(l)} \rangle = \langle k \rangle$, and l is the number of iterations.
- 3) Let $i = j^{(l-1)} + 1, j^{(l)} = 0$.
- 4) Determine whether n_i is less than $\langle k^{(l)} \rangle + 1$. If so, proceed to step 5, otherwise proceed to step 8.
- 5) Let $j^{(l)} = i$.
- 6) Let $i = i + 1$.
- 7) Determine whether i is more than c . If so, proceed to step 8, otherwise return to step 4.
- 8) Determine whether $j^{(l)}$ is more than 0. If so, proceed to step 9, otherwise proceed to step 13.
- 9) Compute the number of remaining lines in the l^{th} iteration:

$$e^{(l)} = e^{(l-1)} - \sum_{m=1}^{j^{(l)}} n_m(n_m - 1)/2 \quad (25)$$

- 10) Compute the number of remaining nodes in the l^{th} iteration:

$$n^{(l)} = n^{(l-1)} - \sum_{m=1}^{j^{(l)}} n_m \quad (26)$$

- 11) Compute the average degree of the remaining part of the graph in the l^{th} iteration:

$$\langle k^{(l)} \rangle = 2e^{(l)}/n^{(l)} \quad (27)$$

- 12) Let $l = l + 1$, then return to step 3.
- 13) Generate complete sub-graphs for the first to $(j^{(l-1)})^{\text{th}}$ connected component.
- 14) Generate random sub-graphs for the $(j^{(l-1)})^{\text{th}}$ to c^{th} connected component with $\langle k^{(l-1)} \rangle$ using the algorithm in section 3.1.2. Then, exit the loop, and the random graph with c components is obtained by putting all of the connected components together.

3.3 Electrical parameters

After the generating the random power grid topology, electrical parameters need to be determined. In this paper, we discuss two issues: bus type assignment and line impedance assignment.

3.3.1 Bus type assignment

In power transmission networks, there are three types of buses: generation (G), load (L) and connection (C). Different power systems have varying bus type ratios. The generation buses may comprise (10-40)% of total grid; the load buses (40-60%); and the connection buses (10-20)% [21]. Random bus type assignment only according to the bus type ratios may lead to unrealistic electrical characteristics. In this paper, we apply the method presented in [22] to assign the bus types in the randomly generated power grid topologies.

3.3.2 Line impedance assignment

As indicated in (7), the node admittance matrix Y can be expressed as a function of the node-branch incidence matrix and the branch impedance matrix. If line impedances are determined, then Y can be derived. The study in [18] showed that the line impedance in the power grid is heavy-tailed and can be captured quite accurately by a clipped double Pareto lognormal (DPLN) distribution.

After the assignment of bus types, we can define six types of lines: GG, GL, GC, LL, LC and CC. For example, GC denotes the line connecting a generation bus and a connection bus, and similarly for the other pairs. Through bus types and the node adjacency matrix, we can determine the type of each line. The distribution of line impedance of each type is not the same. A unique DPLN distribution is not adequate for six types of lines. We will comprehensively discuss this issue in our future work.

4 Experimental studies

The experiments were run on a Core 2 DUO 3.3 GHz CPU machine with 6 GB ROM under Windows 7. All algorithms were implemented in MATLAB R2012b. We measure the execution time as user time, averaging the results over 100 runs.

4.1 Instructions of existing algorithms

Two existing random topology generation algorithms are introduced in this section, consisting of small-world [10] and RT-nested-Smallworld [18].

4.1.1 Small-world networks

Watts–Strogatz small-world models (“small-world” for short) are often used for modeling power grid topologies. The small-world network is generated starting from a



regular ring lattice, then using a small probability p , by rewiring some local links to an arbitrary node chosen uniformly at random in the entire network (to make it a small-world network).

1) Connectivity check

Small-world networks are not always connected. Their connectivity depends on the average degree and the rewiring probability p . Therefore, we need to add a loop to check connectivity each time a small-world network is generated.

2) The rewiring probability

To make the small-world networks match the characteristics of the realistic power grid topologies, we need to choose an appropriate rewiring probability p . In Fig. 1, we present the degree distributions of connected small-world networks with $n = 3000$, $\langle k \rangle = 2, 4$, and $p = 0.1, 0.5, 0.9$. Horizontal and vertical axis represent node degree and probability mass function (i.e. PMF) respectively.

It can be seen that the degree distributions of small-world networks become broad and close to a Poisson distribution, as the value of p increases. In other words, the rewiring probability p determines the degree of randomness of small-world networks—a larger p means a more randomized network.

4.1.2 RT-nested-Smallworld

There is no specific instruction about how to choose the parameters of RT-nested-Smallworld in the literature, so we provide some guidance.

1) Connectivity check

We first review the procedure of the algorithm. The RT-nested-Smallworld network is formed using a hierarchical process: first form connected sub-networks with size limited by the connectivity requirement; then connect the sub-networks through lattice

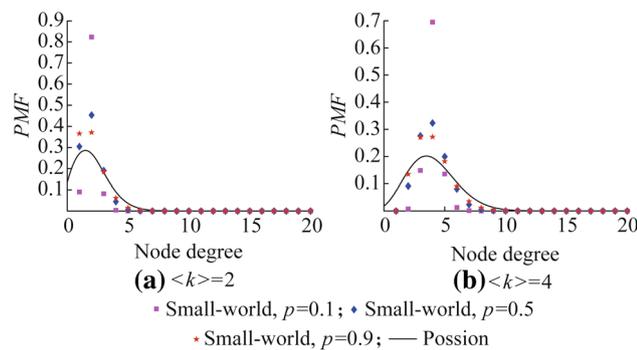


Fig. 1 Degree distributions of small-world networks with different rewiring probability p

connections. To make sure the generated sub-networks are connected, the experiments in [18] have shown that: for $\langle k \rangle = 2$ to 3, the sub-network size should be limited to no greater than 30; and for $\langle k \rangle = 4$ to 5, no greater than 300. Therefore, the first step of this model is to select the size of sub-networks according to the connectivity limitation. In this paper, we call the sub-network “RT-Smallworld”.

2) Parameter selection

There are four parameters to be determined when RT-Smallworld is formed, which are the neighborhood range d , the Markov transition probabilities (α, β) , and the rewiring probability p_{rw} . In the experiment, we investigate four groups of different RT-Smallworld sub-networks: $\{n = 30, \langle k \rangle = 2\}$; $\{n = 30, \langle k \rangle = 3\}$; $\{n = 300, \langle k \rangle = 4\}$; $\{n = 300,$

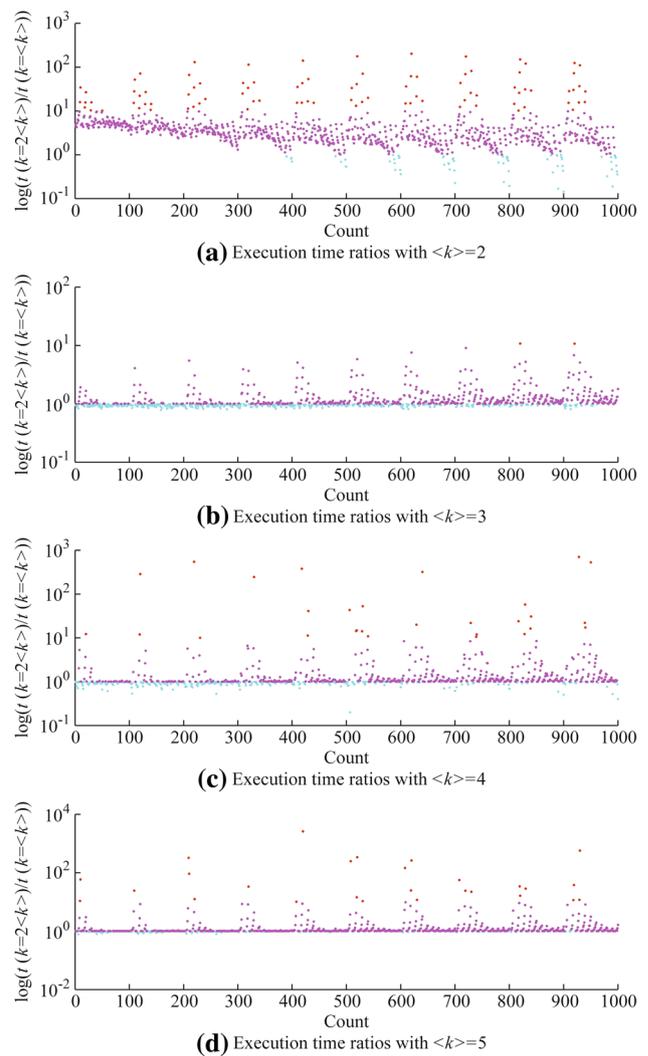


Fig. 2 Execution time ratios between $d = 2\langle k \rangle$ and $d = \langle k \rangle$ for generating RT-Smallworld networks

$\langle k \rangle = 5$. The parameters: $d = \{\langle k \rangle, 2\langle k \rangle\}$; $\alpha, \beta,$ and p_{rw} all range from 0.1 to 1 with step 0.1.

Firstly, we determine the value of d by setting it as the unique variable. The execution time ratios of $d = 2\langle k \rangle$ and $\langle k \rangle$ are recorded, when $\alpha, \beta,$ and p_{rw} remain unchanged. There are 1000 values of time ratio in each

group. The four groups of time ratios are shown in Fig. 2. The red, pink and light blue dots means that the ratio is over 10, between 1 and 10, and less than 1, respectively. Obviously, the algorithm with $d = \langle k \rangle$ is faster than the one with $d = 2\langle k \rangle$. Hence, the value of d is determined to be $\langle k \rangle$.

Secondly, the values of $\alpha, \beta,$ and p_{rw} are to be determined. The definition of Kullback–Leibler divergence was introduced in section 2, and it can be an index to measure the difference between two degree distributions. Based on the D_{KL} , we propose an offline method to determine the parameters $\alpha, \beta,$ and p_{rw} for RT-Smallworld.

In (2), $P(x)$ and $Q(x)$ represent the degree distribution of the realistic power grid topology to be simulated and the

Table 1 Four typical power system cases

Name	IEEE-300	GB-2224	PL-2383	EU-2869
Number of nodes	300	2224	2383	2869
Number of lines	409	2804	2886	3968
Average degree	2.73	2.52	2.42	2.77

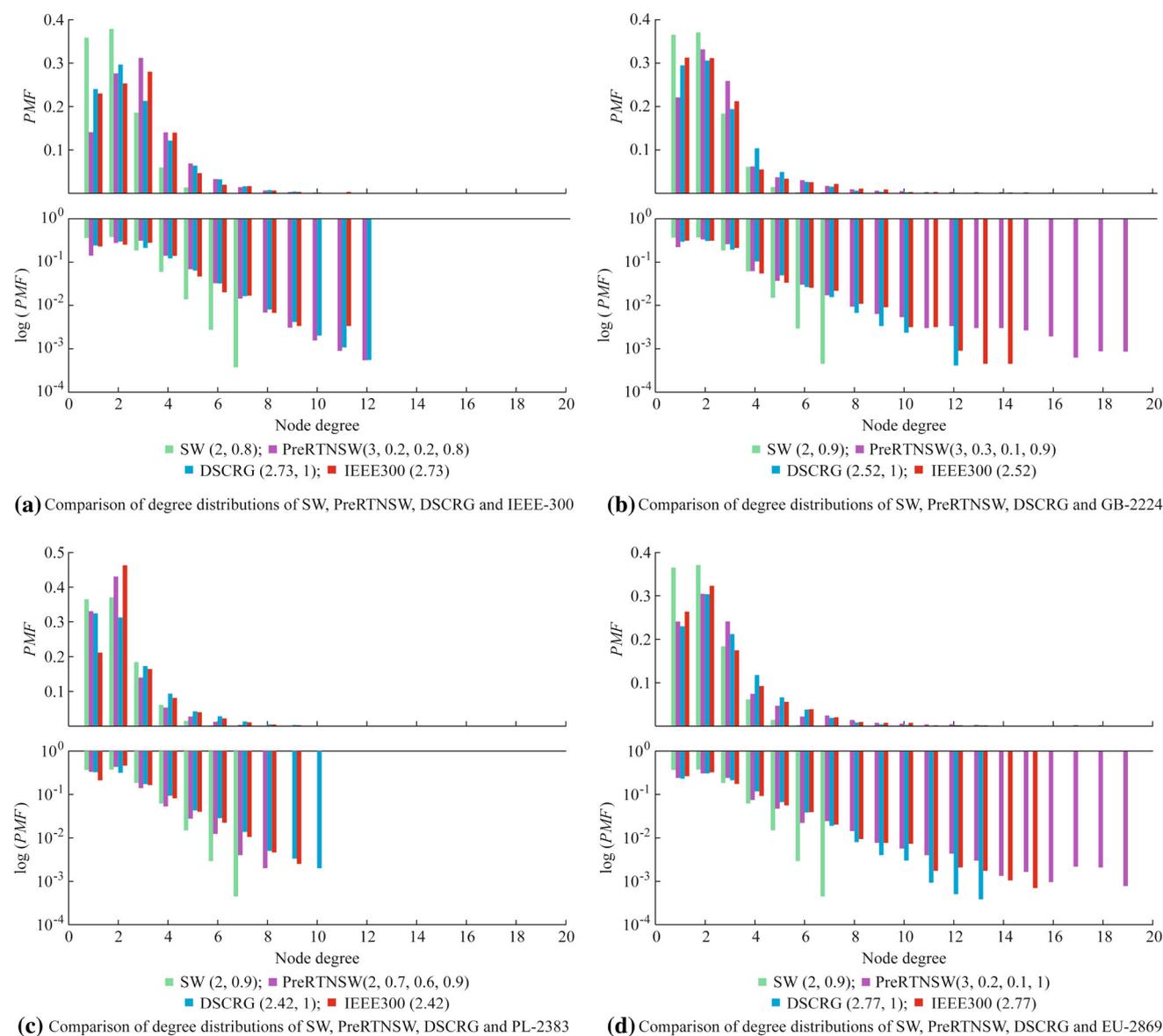


Fig. 3 Comparisons of degree distributions



generated RT-Smallworld network, respectively. The purpose is to select the RT-Smallworld network with the most similar degree distribution to the realistic power grid topology.

For this purpose, we obtain the degree distributions of RT-Smallworld networks under different α , β , and p_{rw} in advance, and use them as an offline database. When the degree distribution of the targeted power grid topology $P(x)$ is given, we compare the values of $D_{KL}(P||Q)$ and choose the RT-Smallworld network with the smallest $D_{KL}(P||Q)$. Then the corresponding values of α , β , and p_{rw} are obtained.

4.2 Typical power system cases

Four typical power system test cases are introduced in this section, which are the IEEE-300 test system [1], the GB-2224 power system [3], the PL-2383 power system [3] and the EU-2869 power system [23]. These power grid topologies are summarised in Table 1 and are all connected.

4.3 Comparison of simulation results

It should be noted that neither small-world nor RT-nested-Smallworld are capable of generating random topologies with multiple connected components. In section 4.3.1 we compare the performance of the algorithms for generating random topologies with only one connected component, and in section 4.3.2 we present the results of

generating random topologies with prescribed numbers of connected components by using DSCRG.

4.3.1 Results of generating random topologies with one connected component

The four typical power system cases introduced in section 4.2 are simulated by the Small-world (SW), the RT-nested-Smallworld based on D_{KL} offline database (PreRTNSW) and the DSCRG algorithms. The comparison results of the three algorithms are presented in Table 2. $\langle k_{real} \rangle$ represents the realistic average degree of each power grid topology, while $\langle k_{ob} \rangle$ is the average degree of the obtained random topology. We introduce an index ε to denote the relative error between $\langle k_{real} \rangle$ and $\langle k_{ob} \rangle$, where $\varepsilon = |\langle k_{real} \rangle - \langle k_{ob} \rangle|$. The smaller ε is, the better is the corresponding algorithm. The numbers in parentheses are the parameters of the corresponding algorithm. For SW, the parameters are $(\langle k \rangle, p)$; for PreRTNSW, $(\langle k \rangle, \alpha, \beta, p_{rw})$; for DSCRG, $(\langle k \rangle, c)$.

There are four columns under each algorithm name, which are execution time, average degree of the generated random topology, relative error ε and D_{KL} of degree distributions between the realistic power grid topology and the generated random topology. There are four different values of n , the first is equal to the node number of the realistic power grid, and the others are 1000, 3000 and 5000. The best performances are listed in bold font.

From the above table, we observe that DSCRG is the fastest algorithm for generating all the random topologies.

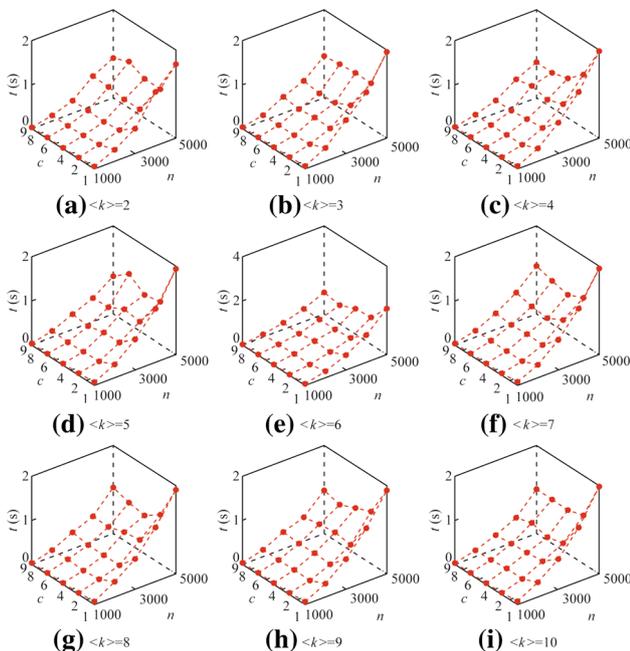


Fig. 4 Execution times of generated random graphs with prescribed numbers of connected components

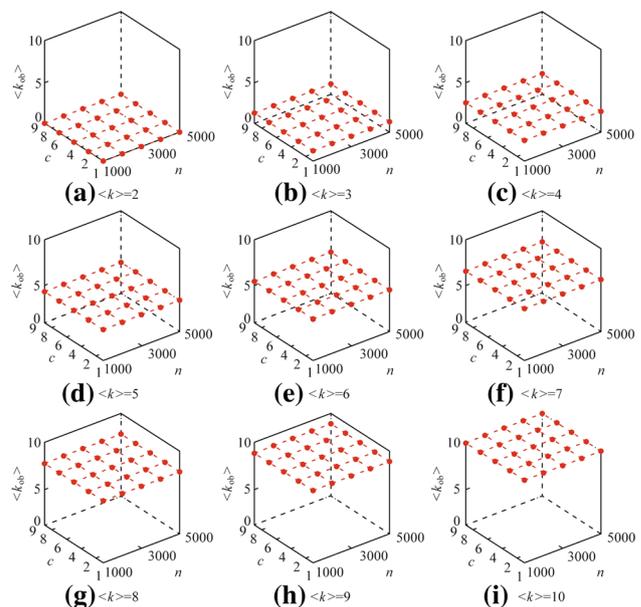


Fig. 5 Average degrees of generated random graphs with prescribed numbers of connected components

Table 2 Simulation results of SW, PreRTNSW and DSCRG for four typical power systems

n	SW (2, 0.8)				PreRTNSW (3, 0.2, 0.2, 0.8)				DSCRG (2.73, 1)			
	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}
IEEE-300, $\langle k_{real} \rangle = 2.73$												
300	0.035	2.00	0.73	0.3092	1.071	2.97	0.24	0.0389	0.005	2.73	0	0.0246
1000	0.507				1.493				0.064			
3000	7.673				2.565				0.657			
5000	24.014				3.654				1.895			
n	SW (2, 0.9)				PreRTNSW (3, 0.3, 0.1, 0.9)				DSCRG (2.52, 1)			
	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}
GB-2224, $\langle k_{real} \rangle = 2.52$												
2224	4.220	2.00	0.52	0.3391	3.869	2.97	0.45	0.0379	0.351	2.52	0	0.0254
1000	0.550				2.087				0.066			
3000	8.432				4.491				0.646			
5000	25.084				6.035				1.903			
n	SW (2, 0.9)				PreRTNSW (2, 0.7, 0.6, 0.9)				DSCRG (2.42, 1)			
	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}
PL-2383, $\langle k_{real} \rangle = 2.42$												
2383	4.700	2.00	0.42	0.1419	14.412	2.08	0.34	0.0853	0.381	2.42	0	0.0587
1000	0.550				7.686				0.065			
3000	8.432				15.958				0.649			
5000	25.084				26.122				1.910			
n	SW (2, 0.9)				PreRTNSW (3, 0.2, 0.1, 1)				DSCRG (2.77, 1)			
	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}	Time(s)	$\langle k_{ob} \rangle$	ε	D_{KL}
EU-2869, $\langle k_{real} \rangle = 2.77$												
2869	8.159	2.00	0.77	0.4996	3.293	2.97	0.20	0.0278	0.599	2.77	0	0.0163
1000	0.550				1.805				0.064			
3000	8.432				3.500				0.648			
5000	25.084				5.206				1.896			

The bold number means the best result in a row

The execution time of DSCRG is only related to n ; for SW and PreRTNSW, it depends on $\langle k \rangle$, and more time is required to check connectivity when $\langle k \rangle$ is smaller.

The obtained average degree $\langle k_{ob} \rangle$ of DSCRG is identical to the realistic average degree $\langle k_{real} \rangle$, because the number of lines is precisely controlled in the second stage of DSCRG. By contrast, $\langle k_{ob} \rangle$ of SW are even numbers, and $\langle k_{ob} \rangle$ of PreRTNSW are near integers, because the process of generating the regular ring lattice has limited the value of $\langle k \rangle$. The values of $\langle k_{real} \rangle$ are between 2 to 3, so the $\langle k \rangle$ of SW can only be 2, and $\langle k \rangle$ of PreRTNSW needs to be rounded to the nearest integer.

The relative error ε shown in Table 2 is the difference value between $\langle k_{real} \rangle$ and $\langle k_{ob} \rangle$. For DSCRG, ε is zero, clearly less than for SW and PreRTNSW. This proves that

the average degree of the generated random topology by DSCRG can be precisely controlled.

The values of D_{KL} of DSCRG are the smallest for all the generated random topologies, which means the degree distributions of DSCRG are most similar to the realistic ones. The values of D_{KL} of PreRTNSW are the second smallest, because of the offline method we proposed for selecting parameters. However, the degree distribution is somehow affected by the average degree, so the D_{KL} of PreRTNSW is greater than that of DSCRG.

We provide the bar charts of the degree distributions in Fig. 3, from which can be seen that SW networks have the greatest difference compared to realistic power grid topologies. As shown in Fig. 3(a), the node proportion of DSCRG is closer to that of IEEE-300 than the node



proportion of PreRTNSW when the node degree is 1, while the opposite is true when the node degree is 3.

Fig. 3(b) shows that the performance of DSCRG is better than that of PreRTNSW when the node degree is from 1 to 3. On the contrary, when the node degree is over 3, the performance of PreRTNSW is slightly better than the one of DSCRG. Some nodes with degree greater than 14 are generated by PreRTNSW, but there is no such node in GB-2224.

Fig. 3(c) shows that the degree distribution of DSCRG is closer with PL-2383 than that of PreRTNSW except when the node degree is 2. Finally, Fig. 3(d) shows that the degree distributions of DSCRG and PreRTNSW are nearly the same when the node degree is below 12. Nodes with degree 14 and 15 are not generated by the former, while nodes with degree greater than 15 (that don't exist in EU-2869) are generated by the latter.

From the above analyses, we conclude that DSCRG is superior to SW and PreRTNSW with respect to execution time, average degree and distribution degree for simulating the realistic power grid topologies.

4.3.2 Results of generating random graphs with prescribed numbers of connected components

An experimental study was conducted to generate random topologies with prescribed numbers of connected components by applying DSCRG only. With $n = \{1000, 2000, 3000, 4000, 5000\}$, $\langle k \rangle = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $c = \{1, 3, 5, 7, 9\}$, $225 (n \times \langle k \rangle \times c)$ kinds of different random graphs were generated. The execution times are shown in Fig. 4. All the random graphs can be generated within 2 seconds. The execution time increases linearly with the graph size. The average degrees are shown in Fig. 5. We observe that the obtained average degrees are the same as the given value of $\langle k \rangle$ for any c and n .

5 Conclusion

Numerous studies on power systems call for the generation of random graphs. In order to substitute generated random graphs for realistic power grid topologies in simulations, we have studied several existing random graph generation algorithms and pointed out their shortcomings clearly. To effectively model power grid topologies, we have presented a novel algorithm named “dual-stage constructed random graph”. In the first stage, a spanning tree with a given number of nodes is randomly constructed; in the second stage, the spanning tree is expanded to the required random graph by arbitrarily adding a certain number of lines to it. To simulate power grid topologies under fault conditions, the algorithm has been generalized

to generate random graphs with prescribed numbers (one or multiple) of connected components. By empirical comparisons, it has been proved that:

- 1) When the number of connected components is prescribed to be one, the proposed DSCRG can guarantee the connectivity of the generated random graph. Unlike the existing algorithms, there is no need to check connectivity repeatedly.
- 2) The number of lines in the generated random graph by DSCRG can be precisely controlled to ensure that there is no difference between the obtained and the realistic average degrees.
- 3) Similar node degree distributions to real-world power grid topologies have been obtained by DSCRG.
- 4) Random graphs with a prescribed number of connected components can be generated by DSCRG within a shorter time frame than existing algorithms.

In general, the proposed DSCRG is a preferable algorithm that is capable of quickly generating a large number of sparse random graphs with prescribed numbers of connected components and the appropriate degree distribution to substitute for real-world power grid topologies.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- [1] UW Electrical Engineering (2016) Power systems test case archive. <http://www.ee.washington.edu/research/pstca>. Accessed 03 May 2016
- [2] Yu YX, Ma SQ (2014) 1-Neighbour knapsack problem and prospective greedy algorithm of intentional islanding in active distribution network. *Sci China Tech Sci* 57(3):568–577
- [3] LIINES (2016) Smart power grid test case repository. <http://amfarid.scripts.mit.edu/Datasets/SPG-Data/index.php>. Accessed 03 May 2016
- [4] Carreras B, Lynch V, Sachtjen M et al (2001) Modeling blackout dynamics in power transmission networks with simple structure. In: Proceedings of the 34th Hawaii international conference on system sciences, Hawaii, USA, 6–6 Jan 2001, 9 pp
- [5] Carreras BA, Lynch VE, Dobson I et al (2002) Critical points and transitions in an electric power transmission model for cascading failure blackouts. *Chaos* 12(4):985–994
- [6] Carreras BA, Lynch VE, Dobson I et al (2004) Complex dynamics of blackouts in power transmission systems. *Chaos* 14(3):643–652
- [7] Parashar M, Thorp JS, Seyler CE (2004) Continuum modeling of electromechanical dynamics in large-scale power systems. *IEEE Trans Circuits Syst I Reg Papers* 51(9):1848–1858

- [8] Pagani GA, Aiello M (2011) The power grid as a complex network: a survey. *Physica A* 392(11):2688–2700
- [9] Meng ZW, Lu ZX, Song JY (2004) Comparison analysis of the small-world topological model of Chinese and American power grids. *Automat Electron Power Syst* 28(15):21–24
- [10] Watts DJ, Strogatz SH (1998) Collective dynamics of ‘small-world’ networks. *Nature* 393(6684):440–442
- [11] Pagani GA, Aiello M (2011) Towards decentralization: a topological investigation of the medium and low voltage grids. *IEEE Trans Smart Grid* 2(3):538–547
- [12] Wang Z, Scaglione A, Thomas RJ (2010) The node degree distribution in power grid and its topology robustness under random and selective node removals. In: Proceedings of the 2010 IEEE international conference on communications workshops, Capetown, South Africa, 23–27 May 2010, 5 pp
- [13] Erdős P, Rényi A (1959) On random graphs I. *Publ Math Debrecen* 6:290–297
- [14] Batagelj V, Brandes U (2005) Efficient generation of large random networks. *Phys Rev E* 71(3):036113
- [15] Nobari S, Lu X, Karras P et al (2011) Fast random graph generation. In: Proceedings of the 14th international conference on extending database technology, Uppsala, Sweden, 21–24 Mar 2011, 12 pp
- [16] Sachtjen ML, Carreras BA, Lynch VE (2000) Disturbances in a power transmission system. *Phys Rev E* 61(5):4877–4882
- [17] Wang Z, Thomas RJ, Scaglione A (2008) Generating random topology power grids. In: Proceedings of 41st Hawaii international conference on system sciences, Hawaii, USA, 7–10 Jan 2008, 9 pp
- [18] Wang Z, Scaglione A, Thomas RJ (2010) Generating statistically correct random topologies for testing smart grid communication and control networks. *IEEE Trans Smart Grid* 1(1):28–39
- [19] Bollobás B (1981) Degree sequences of random graphs. *Discrete Math* 33(1):1–19
- [20] Kullback S, Leibler RA (1951) On information and sufficiency. *Ann Math Stat* 22(22):79–86
- [21] Wang Z, Thomas RJ (2015) On bus type assignments in random topology power grid models. In: Proceedings of 48 annual Hawaii international conference on system science, Hawaii, USA, 5–8 Jan 2015, 10 pp
- [22] Elyas SH, Wang Z (2016) A multi-objective optimization algorithm for bus type assignments in random topology power grid model. In: Proceedings of 48 annual Hawaii international conference on system science, Hawaii, USA, 5–8 Jan 2016, 10 pp
- [23] Fliscounakis S, Panciatici P, Capitanescu F et al (2013) Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive and corrective actions. *IEEE Trans Power Syst* 28(28):4909–4917

Shiqian MA received his Ph.D. degree in electrical engineering from Tianjin University, China in 2016. He is currently working in Electric Power Research Institute of State Grid Tianjin Electric Power Company, China. His research interests include smart grid, energy internet and graph theory.

Yixin YU received the B.E. and M.E. degree from Tianjin University, China, in 1958 and 1963, respectively. He was elected to the Chinese Academy of Engineering in 2005. He is currently a professor of school of electrical and information engineering at Tianjin University, China. His technical background is broadly in the areas of system analysis, simulation, and planning of power systems. He is dedicated to promoting smart grid and distributed generation in China. He has published in the areas of stability and security of power transmission systems, as well as planning, operation and management of distribution systems.

Lei ZHAO received the B.E. and M.E. degree from Tianjin University, China, in 2013 and 2016, respectively. He is currently working in State Grid Tianjin Electric Power Company, China. His research interests include application of the graph theory in power grids and AMI in smart grid.

