

Research Article

A Weighted Multi-Local-World Network Evolving Model and Its Application in Software Network Modeling

Zengyang Li ^{1,2}, Hui Liu ³, Jun-An Lu,^{2,4} and Bing Li ^{1,2}

¹School of Computer Science, Wuhan University, Wuhan 430072, China

²Complex Networks Research Center, Wuhan University, Wuhan 430072, China

³School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

⁴School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Bing Li; bingli@whu.edu.cn

Received 21 February 2018; Accepted 16 April 2018; Published 16 May 2018

Academic Editor: Michele Baccocchi

Copyright © 2018 Zengyang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The phenomenon of local worlds (also known as communities) exists in numerous real-life networks, for example, computer networks and social networks. We proposed the Weighted Multi-Local-World (WMLW) network evolving model, taking into account (1) the dense links between nodes in a local world, (2) the sparse links between nodes from different local worlds, and (3) the different importance between intra-local-world links and inter-local-world links. On topology evolving, new links between existing local worlds and new local worlds are added to the network, while new nodes and links are added to existing local worlds. On weighting mechanism, weight of links in a local world and weight of links between different local worlds are endowed different meanings. It is theoretically proven that the strength distribution of the generated network by the WMLW model yields to a power-law distribution. Simulations show the correctness of the theoretical results. Meanwhile, the degree distribution also follows a power-law distribution. Analysis and simulation results show that the proposed WMLW model can be used to model the evolution of class diagrams of software systems.

1. Introduction

Networked structures appear in a wide range of complex systems, such as the Internet, traffic systems, and scientific collaboration relationships. Since the small-world and scale-free network models were proposed [1, 2], researchers have made tremendous progress in network modeling [3–5].

In 2003, Li and Chen proposed a local-world evolving network model, which represents a transition between power-law and exponential scaling [6]. The concept of local-world structure exists in various real-life complex networks and has been taken by many studies [7, 8], allowing better describing and understanding more real-life complex networks. Based on the aforementioned local-world evolving network model, Pan et al. proposed two generalized local-world models for unweighted and weighted complex networks, respectively [8]. Li et al. proposed a weighted local-world evolving network model based on the edge weights preferential selection mechanism [9]. However, in the

mentioned models, the nodes composing a local world were selected randomly from the network and the number of nodes in a local world is fixed, which did not comply well with the real-life complex networks.

In 2005, Chen et al. proposed a Multi-Local-World (MLW) model [10] which aims at modeling the Internet and emphasizes the property of the Multi-Local-World of the structure of the Internet. A comparison clearly demonstrated that the MLW model was the best model for describing the Internet topology in the sense that the basic network properties (e.g., mean degree and clustering coefficient) of the MLW model are closer to those of the real Internet topology than other models, such as the BA model. In the MLW model, local worlds are relatively stable in the sense that each local world is not comprised of randomly selected nodes from the network when new nodes are added to a local world of the network.

The aforementioned network models did not take into account the difference between a link within a local world and

a link across two local worlds. In real-life cases, an inter-local-world link tends to play a more important role than an intra-local-world link. For instance, in a computer network, a link between two nodes from two subnetworks is responsible for the communication between the two subnetworks. In addition, nodes differ on their capacity, which can be represented by their strengths.

Inspired by the MLW model, we try to extend the MLW model and propose a Weighted Multi-Local-World (WMLW) model in which a weight mechanism is introduced. To our knowledge, there has been no study about MLW taking into account the edge weight and node strength. In fact, the real-life complex systems would be better reflected if the edge weight and node strength were taken into consideration. The proposed WMLW model does not aim at modeling the Internet, but it should be feasible to model more real-life complex networks, for example, software systems.

There have been some studies on diverse aspects of software systems from a perspective of complex networks. Qu et al. explored community structure of software call graph and its applications in class cohesion measurement [11]. Concas et al. investigated software quality and community structure in Java software networks [12]. A complex network approach was used to study software dependency network evolution [13, 14]. Chong and Lee used weighted complex network with graph theory analysis to automate the derivation of clustering constraints from object-oriented software [15]. Joblin et al. investigated the evolutionary trends of developer coordination using a network approach [16]. Ding et al. identified key classes in weighted software networks [17]. In this paper, we apply the MLWM to modeling the evolution of object-oriented software networks, in which a node is a class and an edge between two nodes is a dependency link between two classes.

The remaining of this paper is organized as follows. We propose the WMLW model in Section 2. In Section 3, we analytically study the strength distribution of WMLW model. We report numerical simulations in Section 4 and the application of WMLW model to software systems in Section 5. Finally, we conclude our work with future work in Section 6.

2. Weighted Multi-Local-World Model

We first introduce several key concepts on the WMLW model and then propose two preferential attachment rules to construct the WMLW model and finally describe the WMLW model in detail.

2.1. Key Concepts on the WMLW Model. A weighted network is characterized by an adjacency matrix W , whose element w_{ij} denotes the weight of the edge between nodes i and j , where $i, j = 1, 2, \dots, N$, and N is the total number of nodes of the network.

Here, we restrict our interest to undirected networks in which weights of edges are symmetric ($w_{ij} = w_{ji}$) and assume that $w_{ii} = 0$. In our model, when adding new edge connected nodes i and j that belong to the same local world, we have

$w_{ij} = a$; otherwise, when adding a new edge between nodes i and j from different local worlds, we have $w_{ij} = b$.

Naturally, as the generalization of degree k_i of node i , we define the strength of node i as $s_i = \sum_{j \in \Gamma(i)} w_{ij}$, where $\Gamma(i)$ denotes the set of neighbors of node i . The strength of a node integrates the information about its connectivity and the weights of its links [18, 19]. The strength of the node reflects its power and importance. A larger strength of a node indicates that it is more important.

2.2. Preferential Attachment Rules

2.2.1. Probability about Selecting a Local World following the Scale Preferential Rule. The scale of local world Ω is denoted by M_Ω ; the total number of local worlds is denoted by N_c . K is a constant that represents the ‘‘attractiveness’’ of local world Ω , and it is used to govern the probability for those ‘‘young’’ local worlds to get new links [10]. In the WMLW model, the probability with which local world Ω is selected is described by

$$\Pi_{\text{local}}(\Omega) = \frac{M_\Omega + K}{\sum_{j=1}^{N_c} (M_j + K)}. \quad (1)$$

If the constant $K = 0$, we have

$$\Pi_{\text{local}}(\Omega) = \frac{M_\Omega}{\sum_{j=1}^{N_c} M_j}. \quad (2)$$

For the sake of simplicity, we consider $K = 0$ (i.e., (2)) in the rest of this paper.

In (2), $\sum_{j=1}^{N_c} M_j$ denotes the sum of the scales of local worlds in the network, that is, the total number of the nodes in the network.

In fact, it is more reasonable to select a local world following the scale preferential rule than to select a local world randomly. In many real cases, the scales of different local world are not equal, for example, software systems, Economic Trade Web, and Internet. The larger the sale of a local world is, the more attractive it is; that is, it also obeys the rule of ‘‘the rich get richer.’’

2.2.2. Probability of Selecting a Node within a Local World following the Strength Preferential Rule. The probability that node i in local world Ω is chosen is

$$\prod_{i \in \Omega} (i) = \frac{s_i}{\sum_{j \in \Omega} s_j}. \quad (3)$$

2.3. Algorithm of the WMLW Model

Step 1 (initial condition). Start with an initial network containing n_1 local worlds, in which it is supposed that there are m_1 nodes and e_1 links in each local world, and every edge is assigned a weight a .

Step 2. At each step, perform one of the following four operations on probability:

(i) With probability p_1 , a new local world is created, which contains m_1 nodes and e_1 links.

(ii) With probability p_2 , a new node is added to an existing local world, which has m_2 links connecting to the nodes within the same local world. The local world is selected with a probability given by (2), and then a node in the local world is chosen with a probability given by (3). Every new added edge is assigned a weight a . This process is repeated m_2 times.

(iii) With probability p_3 , m_3 new links are added to a chosen local world. A local world is selected with a probability given by (2), and then one end of a link is selected with a probability given by (3), while the other end of the link is chosen randomly. Every new added edge is assigned a weight a . This process is repeated m_3 times.

(iv) With probability p_4 , a local world is selected and it has m_4 new links to the other existing local worlds. A local world (Ω_1) is selected with a probability given by (2). Then, each link is added according to the following process: a node is selected from Ω_1 with a probability given by (3), and this node acts as one end of a link; the other end of the link is a node selected with a probability given by (3) from another local world (Ω_2) chosen at random; and every new added edge is assigned a weight b . This process is repeated m_4 times.

Step 3. Repeat Step 2 until the total number of the nodes reaches the number given in advance.

In this model, neither a node is allowed linking with itself nor two nodes are allowed linking repeatedly. In addition, the parameters (probabilities) have to satisfy $0 \leq p_1, p_2, p_3, p_4 \leq 1$ and $p_1 + p_2 + p_3 + p_4 = 1$.

3. Strength Distribution of the WMLW Model

Using the mean-field theory [21, 22], one can obtain the strength distribution of node i in local world Ω , which can be divided analytically as follows.

(i) With probability p_1 , create a new local world. In this case, the strength of node i in an existing local world Ω does not change over time, since the original nodes in the newly created local world have no links with any other nodes in existing local worlds. Thus,

$$\left(\frac{\partial S_i}{\partial t}\right)_{(i)} = 0. \quad (4)$$

(ii) With probability p_2 , a new node is added to an existing local world Ω , and this node has m_2 links connected to the nodes within the same local world. Every newly added edge is assigned a weight a :

$$\left(\frac{\partial S_i}{\partial t}\right)_{(ii)} = ap_2 m_2 \cdot \frac{M_\Omega}{\sum_{j=1}^{N_c} M_j} \cdot \frac{s_i}{s_\Omega}. \quad (5)$$

$M_\Omega / \sum_{j=1}^{N_c} M_j$ is the probability that local world Ω is selected according to preferential choosing with the probability given by (2). s_i / s_Ω is the probability that a node is selected according to strength preferential attachment with the probability given by (3). s_Ω represents the total strength of all nodes in local world Ω . There are m_2 links between the newly added node and existing nodes in local world Ω , and the weight of each

newly added edge is a , and thus the coefficient is equal to $ap_2 m_2$.

(iii) With probability p_3 , m_3 links are added to a chosen local world Ω . Each newly added edge is assigned a weight a . We have

$$\left(\frac{\partial S_i}{\partial t}\right)_{(iii)} = ap_3 m_3 \cdot \frac{M_\Omega}{\sum_{j=1}^{N_c} M_j} \cdot \left(\frac{s_i}{s_\Omega} + \frac{1}{M_\Omega}\right). \quad (6)$$

On the right-hand side of (6), local world Ω is selected according to the probability given by (2), s_i / s_Ω represents the preferential selection within local world Ω , and $1 / M_\Omega$ means the random selection of node i within the same local world Ω .

(iv) With probability p_4 , a selected local world (Ω) has m_4 links connected to other existing local worlds. Each newly added edge is assigned a weight b . Hence,

$$\left(\frac{\partial S_i}{\partial t}\right)_{(iv)} = bp_4 m_4 \cdot \frac{M_\Omega}{\sum_{j=1}^{N_c} M_j} \cdot \frac{s_i}{s_\Omega} + bp_4 m_4 \cdot \frac{1}{n_1 + p_1 t} \cdot \frac{s_i}{s_{\text{random-local}}}. \quad (7)$$

In the first term on the right-hand side of (7), $M_\Omega / \sum_{j=1}^{N_c} M_j$ is the probability given by (3) with which we select local world Ω . In the second term on the right-hand side of (7), another local world is chosen at random (i.e., with the probability of $1 / (n_1 + p_1 t)$), and $s_{\text{random-local}}$ represents the total strength of all nodes in the randomly chosen local world. The two ends of a link are all selected by (3) in their own local worlds.

Combining (4)–(7) together, we have

$$\begin{aligned} \frac{\partial s_i}{\partial t} = & (ap_2 m_2 + ap_3 m_3 + bp_4 m_4) \cdot \frac{M_\Omega}{\sum_{j=1}^{N_c} M_j} \cdot \frac{s_i}{s_\Omega} \\ & + ap_3 m_3 \cdot \frac{M_\Omega}{\sum_{j=1}^{N_c} M_j} \cdot \frac{1}{M_\Omega} + bp_4 m_4 \cdot \frac{1}{n_1 + p_1 t} \\ & \cdot \frac{s_i}{s_{\text{random-local}}}. \end{aligned} \quad (8)$$

In the following text, $\langle s \rangle_{\Omega\text{-local}}$ denotes the average strength of the nodes in local world Ω , and $\langle s \rangle$ represents the average strength of all nodes in the network. Hence, we have $s_\Omega = M_\Omega \langle s \rangle_{\Omega\text{-local}}$. Since $s_{\text{random-local}}$ denotes the sum of the strength of all nodes in a local world, which is selected randomly, in the average sense, we have

$$s_{\text{random-local}} = \frac{[n_1 m_1 + (p_1 m_1 + p_2) t] \langle s \rangle}{(n_1 + p_1 t)}. \quad (9)$$

Thus, (8) becomes

$$\begin{aligned} \frac{\partial s_i}{\partial t} = & (ap_2 m_2 + ap_3 m_3 + bp_4 m_4) \\ & \cdot \frac{M_\Omega}{n_1 m_1 + (p_1 m_1 + p_2) t} \cdot \frac{s_i}{M_\Omega \langle s \rangle_{\Omega\text{-local}}} \end{aligned}$$

$$+ ap_3 m_3 \cdot \frac{1}{n_1 m_1 + (p_1 m_1 + p_2) t} + bp_4 m_4 \cdot \frac{s_i}{[n_1 m_1 + (p_1 m_1 + p_2) t] \langle s \rangle}. \quad (10)$$

For large t , we have $n_1 m_1 + (p_1 m_1 + p_2) t \approx (p_1 m_1 + p_2) t$; therefore,

$$\begin{aligned} \frac{\partial s_i}{\partial t} &= (ap_2 m_2 + ap_3 m_3 + bp_4 m_4) \cdot \frac{M_\Omega}{(p_1 m_1 + p_2) t} \\ &\cdot \frac{s_i}{\langle s \rangle_{\Omega\text{-local}}} + ap_3 m_3 \cdot \frac{1}{(p_1 m_1 + p_2) t} + bp_4 m_4 \\ &\cdot \frac{s_i}{(p_1 m_1 + p_2) t \langle s \rangle}. \end{aligned} \quad (11)$$

In the following text, we deduce the strength distribution of the network when the probability $p_1 = 0$ (i.e., in the case when the number of the local worlds is unchanged). In the case when the probability $p_1 \neq 0$, we only give a simulation result for simplicity in Section 4.

When the probability $p_1 = 0$, the number of the local worlds is the initial n_1 . All local worlds perform the same steps (ii), (iii), and (iv), though with different probabilities. Different probabilities just bring on different scales of the local worlds.

We can see from works [6–8] that the average degree of the network is free from the scale of the network. In our model, we can also consider that the average strength of every local world is equal for large t ; that is,

$$\begin{aligned} \langle s \rangle &= \frac{\text{total strength of total nodes}}{\text{the number of total nodes}} \\ &= \frac{\text{const} + 2ap_1 m_1 t + 2ap_2 m_2 t + 2ap_3 m_3 t + 2bp_4 m_4 t}{n_1 m_1 + (p_1 m_1 + p_2) t} \\ &\approx \frac{2ap_1 m_1 + 2ap_2 m_2 + 2ap_3 m_3 + 2bp_4 m_4}{p_1 m_1 + p_2} \end{aligned} \quad (12)$$

(for large t).

Also, $\langle s \rangle_{\Omega\text{-local}}$ is equal to $\langle s \rangle$. Both of them are free from t .

Considering (12), when the probability $p_1 = 0$, (11) becomes

$$\frac{\partial s_i}{\partial t} = \frac{ap_2 m_2 + ap_3 m_3 + 2bp_4 m_4}{2(ap_2 m_2 + ap_3 m_3 + bp_4 m_4)} \cdot \frac{s_i}{t} + \frac{ap_3 m_3}{p_2} \cdot \frac{1}{t}. \quad (13)$$

Define $A = (ap_2 m_2 + ap_3 m_3 + 2bp_4 m_4)/2(ap_2 m_2 + ap_3 m_3 + bp_4 m_4)$ and $B = ap_3 m_3/p_2$.

Therefore,

$$\frac{\partial s_i}{\partial t} = A \frac{s_i}{t} + \frac{B}{t}. \quad (14)$$

By integrating, we have $s_i(t) = \lambda t^A - B/A$, where λ is any constant.

Using the initial condition $s_i(t_i) = m_2$, one obtains

$$s_i(t) = \left(m_2 + \frac{B}{A}\right) \left(\frac{t}{t_i}\right)^A - \frac{B}{A}. \quad (15)$$

Thus,

$$P(s_i(t) < s) = P\left(t_i > t \left(\frac{m_2 + B/A}{s + B/A}\right)^{1/A}\right). \quad (16)$$

In the process of the growth of the network, we add nodes every time interval. Then, the probability density of t_i is

$$p(t_i) = \frac{1}{n_1 m_1 + t}. \quad (17)$$

So

$$\begin{aligned} P\left(t_i > t \left(\frac{m_2 + B/A}{s + B/A}\right)^{1/A}\right) \\ = 1 - \frac{t}{n_1 m_1 + t} \left(\frac{m_2 + B/A}{s + B/A}\right)^{1/A}. \end{aligned} \quad (18)$$

Therefore,

$$p(s) = \frac{\partial P(s_i < s)}{\partial s} = \frac{t}{n_1 m_1 + t} \cdot \frac{(m_2 + B/A)^{1/A}}{A(s + B/A)^{1+1/A}}. \quad (19)$$

For large t , the strength distribution approximately is

$$p(s) \approx \frac{(m_2 + B/A)^{1/A}}{A(s + B/A)^{1+1/A}}, \quad (20)$$

where $A = (ap_2 m_2 + ap_3 m_3 + 2bp_4 m_4)/2(ap_2 m_2 + ap_3 m_3 + bp_4 m_4)$ and $B = ap_3 m_3/p_2$, as defined before.

4. Numerical Simulations

We design several numerical simulations to verify our theoretical analysis results. We consider two cases, that is, a given number of local worlds ($p_1 = 0$) and an increasing number of local worlds ($p_1 \neq 0$).

4.1. A Given Number of Local Worlds ($p_1 = 0$). To testify the theoretical result of strength distribution of the WMLW model, we conducted a numerical simulation in which a network was generated by our WMLW model and the simulation and theoretical results were compared. Figure 1 shows the simulation result. In this simulation, the size of the generated network was set to one million. The details of parameter settings of the WMLW model for the simulation are described in Figure 1. In Figure 1(a), the simulation and theoretical results are shown to facilitate the comparison between them. The theoretical data are calculated by (20), and the simulation data were calculated based on the generated network. The comparison shows that the simulation data and the theoretical data fit well, which implies the correctness of the theoretic result of the strength distribution of the WMLW model. In addition, Figure 1(b) shows that the degree

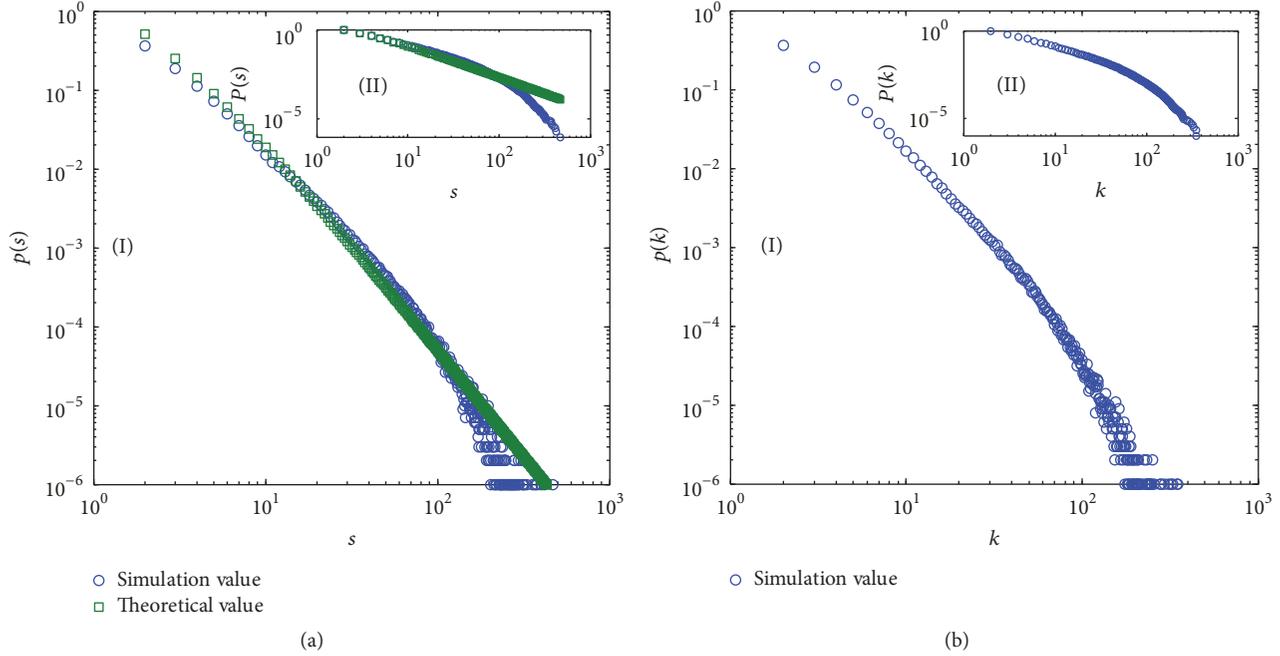


FIGURE 1: Strength and degree distributions of a generated network by the WMLW model. (a) and (b) show the strength distribution and degree distribution of the generated network, respectively. The numerical simulation was carried out under the following settings: $N = 1,000,000$, $n_1 = 100$, $m_1 = 10$, $m_2 = 2$, $m_3 = 2$, $m_4 = 2$, $p_1 = 0.000$, $p_2 = 0.700$, $p_3 = 0.290$, $p_4 = 0.010$, $a = 1$, and $b = 10$.

distribution of the WMLW model also follows a power-law distribution. Hence, the WMLW network is scale-free network.

There is a deviation between the numerical values and theoretical ones for large nodal strengths in Figure 1(a). This is the so-called “fat tail” phenomenon. The theoretical values are calculated when the number of nodes of the network is infinite. In contrast, for a numerical simulation, the number of nodes of the generated network is finite (one million in this simulation), and thus the phenomenon of “fat tail” happens.

Next, we changed the parameters of the WMLW model and calculated the cumulative strength and degree distributions of generated networks. Figure 2 shows the cumulative strength and degree distributions when the size of the generated networks is set to 100,000 and b is set to 5, 10, and 20, respectively. The results show that both the cumulative strength and degree distributions follow power laws.

4.2. An Increasing Number of Local Worlds ($p_1 \neq 0$). We carried out simulations when $p_1 \neq 0$, and the results are shown in Figure 3. The cumulative strength and degree distributions follow power laws. Comparing Figure 3 (the case when $p_1 \neq 0$) with Figure 2 (the case when $p_1 = 0$), not only their cumulative strength distributions but also their degree distributions are very similar. Hence, when the parameter p_1 is small (approximately equal to 0), we consider that $\langle s \rangle_{\Omega\text{-local}}$ is close to $\langle s \rangle$. Therefore, by the same method as before, we have

$$p(s) \approx \frac{(m_2 + B/A)^{1/A}}{A(s + B/A)^{1+1/A}}, \quad (21)$$

where $A = (ap_2m_2 + ap_3m_3 + 2bp_4m_4)/2(ap_2m_2 + ap_3m_3 + bp_4m_4)$ and $B = ap_3m_3/p_2$.

5. Application of the WMLW Model in Software Network Modeling

Object-oriented software systems can be described using class diagrams that depict the relationship between classes. Valverde and Solé found that the small-world and scale-free characteristics also exist in class diagrams of software systems when class diagrams are considered as undirected networks [20]. A class diagram can be considered as a network, in which a node is a class and relationships between classes are links between nodes. In this section, we will employ the WMLW model to model the class diagrams of a set of software systems.

In the design of object-oriented software systems, a software system can be decoupled into components. Classes in the same component tend to own close association and different components couple incompletely because they communicate only by a few interfaces. In fact, this is the rule “high cohesion, low coupling” in the software engineering domain. This rule is clearly implemented in the WMLW model. Thus, the WMLW model is suitable for the modeling of the evolution of class diagrams.

For a software system of a relatively small size, let $p_1 = 0$, which represents that the total number of local worlds is a constant; that is, the architecture is designed in advance and no important function was ignored before implementation. Thus, the number of components (e.g., packages) will not increase. For a software system of a relatively large size

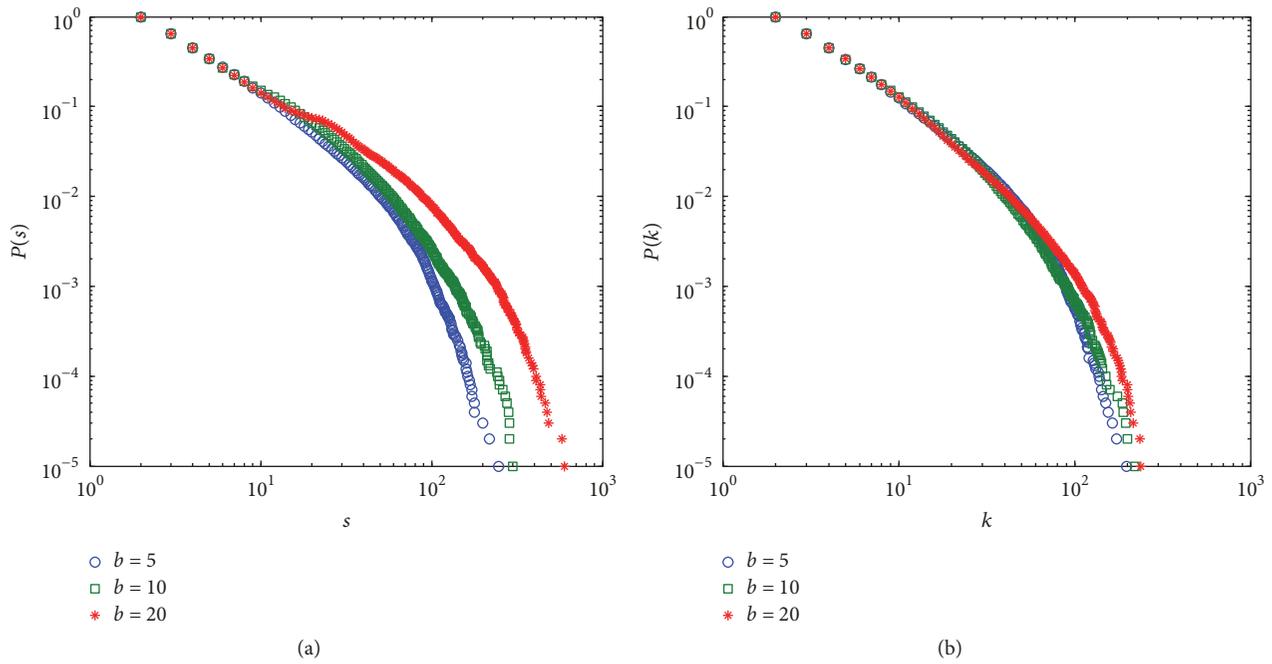


FIGURE 2: Cumulative strength and degree distributions of generated networks by the WMLW model when $p_1 = 0$. (a) and (b) show the cumulative strength distributions and cumulative degree distributions of the generated networks, respectively. The numerical simulations were carried out under the following settings: $N = 100,000$, $n_1 = 30$, $m_1 = 6$, $m_2 = 2$, $m_3 = 2$, $m_4 = 2$, $p_1 = 0.000$, $p_2 = 0.700$, $p_3 = 0.280$, $p_4 = 0.020$, and $a = 1$, when b is set to 5, 10, and 20.

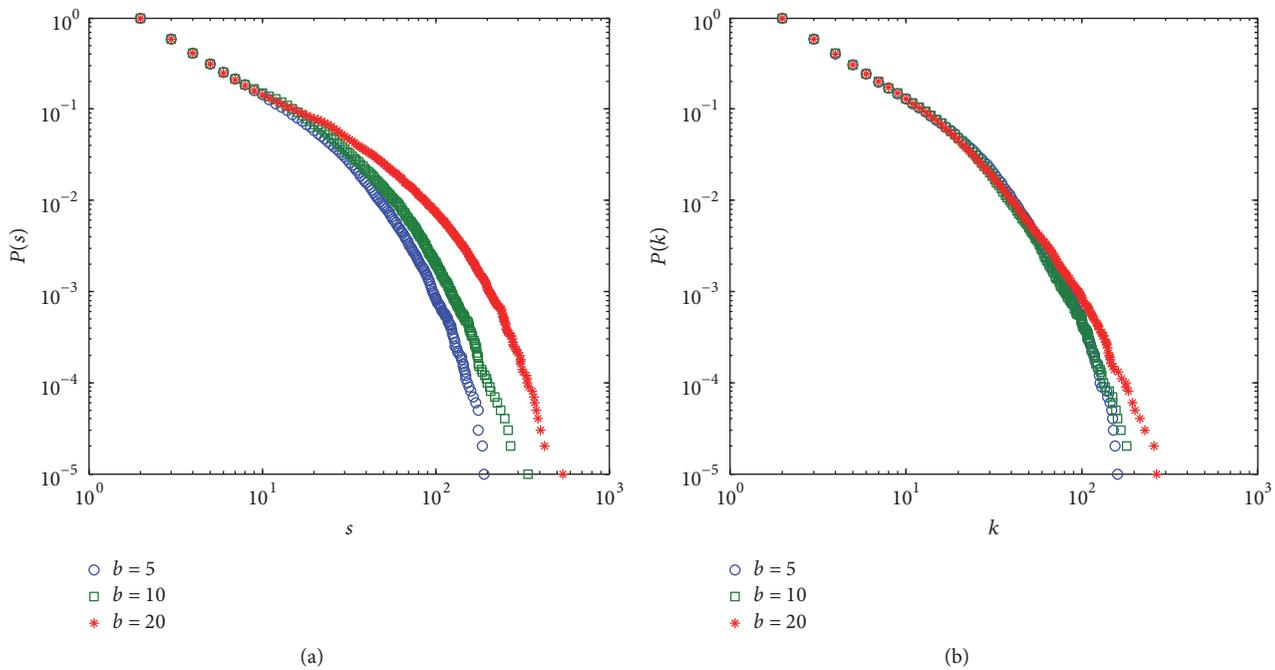


FIGURE 3: Cumulative strength and degree distributions of generated networks by the WMLW model when $p_1 \neq 0$. (a) and (b) show the cumulative strength distributions and cumulative degree distributions of the generated networks, respectively. The numerical simulations were carried out under the following settings: $N = 100,000$, $n_1 = 30$, $m_1 = 6$, $m_2 = 2$, $m_3 = 2$, $m_4 = 2$, $p_1 = 0.005$, $p_2 = 0.695$, $p_3 = 0.280$, $p_4 = 0.020$, and $a = 1$, when b is set to 5, 10, and 20.

TABLE 1: Network properties of the software systems in real and modeling cases.

Software	N_r/N_m	L_r	L_m	P_r	P_m	C_r	C_m
Prospectus	99	168	159	3.80	3.92	0.140	0.145
eMule	129	218	223	3.87	4.07	0.237	0.225
Blender	495	834	856	6.54	6.39	0.155	0.157
OIV	1214	3903	4008	3.99	3.97	0.122	0.130
JDK-A	1376	2162	2314	5.40	5.73	0.159	0.156
CS	1488	3526	3746	3.92	3.94	0.135	0.131

Note. N is the total number of nodes, L is the number of the edges, P is average path length, and C is clustering coefficient. The subscript r represents the real case, and the subscript m represents the modeling case. The real data were reported in [20].

TABLE 2: Parameters of the WMLW model for the modeling of the software systems.

Software	n_1	m_1	m_2	m_3	m_4	p_1	p_2	p_3	p_4	a	b
Prospectus	3	6	1	2	1	0.000	0.700	0.250	0.050	1	10
eMule	4	6	1	3	1	0.000	0.720	0.230	0.050	1	10
Blender	15	6	1	2	1	0.000	0.700	0.260	0.040	1	10
OIV	6	15	3	2	1	0.004	0.696	0.150	0.150	1	10
JDK-A	8	6	1	2	1	0.022	0.680	0.240	0.058	1	10
CS	8	6	1	3	2	0.005	0.595	0.200	0.200	1	10

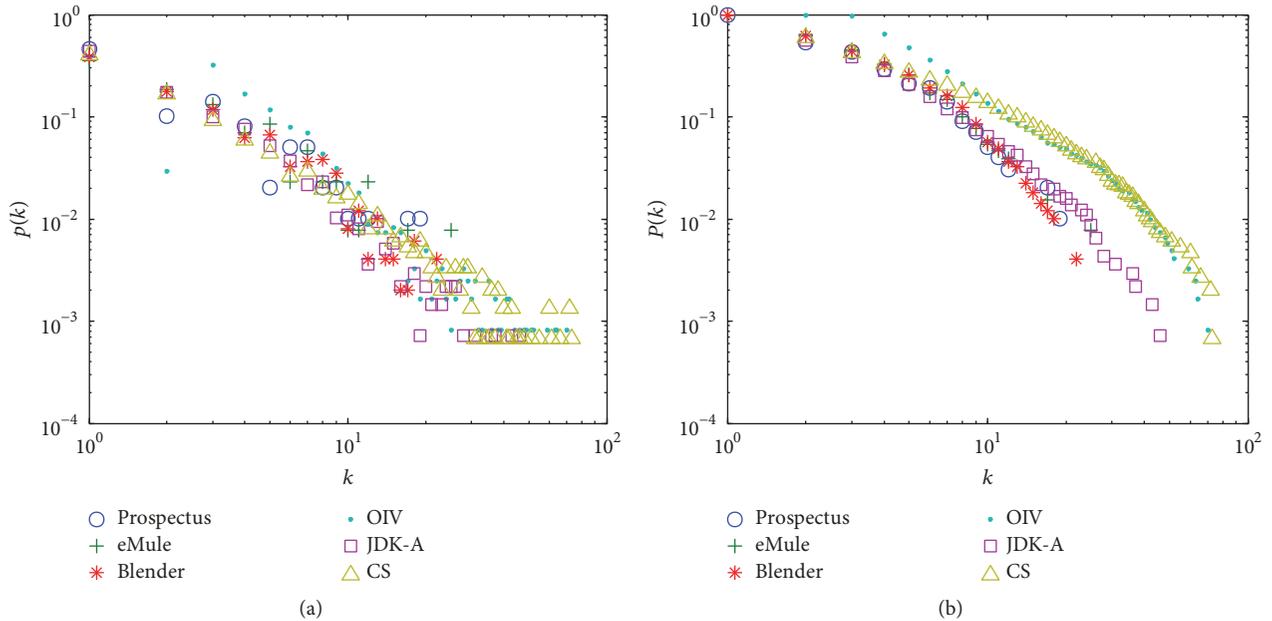


FIGURE 4: Degree distributions and cumulative degree distributions of the modeled software systems. (a) and (b) show the degree distributions and cumulative degree distributions, respectively.

(e.g., more than 1,000 classes), p_1 is assigned a small value, which represents the fact that the total number of local worlds of the software system increases slowly. It is reasonable since adding a component is a big change to the software system at the architecture level, and such changes tend not to happen frequently in the development lifecycle of the software system.

We apply the WMLW model to generating networks with similar network properties to real-life software systems. We get the results shown in Table 1 by regulating the parameters

of the WMLW model to model different software systems. Table 1 shows that the modeling of the software systems is satisfactory, which indicates the success of the application in class diagrams modeling. Table 2 shows the parameters used by the WMLW model in the modeling of each software system. Figure 4 shows the cumulative degree distributions and the cumulative strength distributions of the software systems using the WMLW model. Figure 5 shows the strength distributions and the cumulative strength distributions of the software systems using the WMLW model.

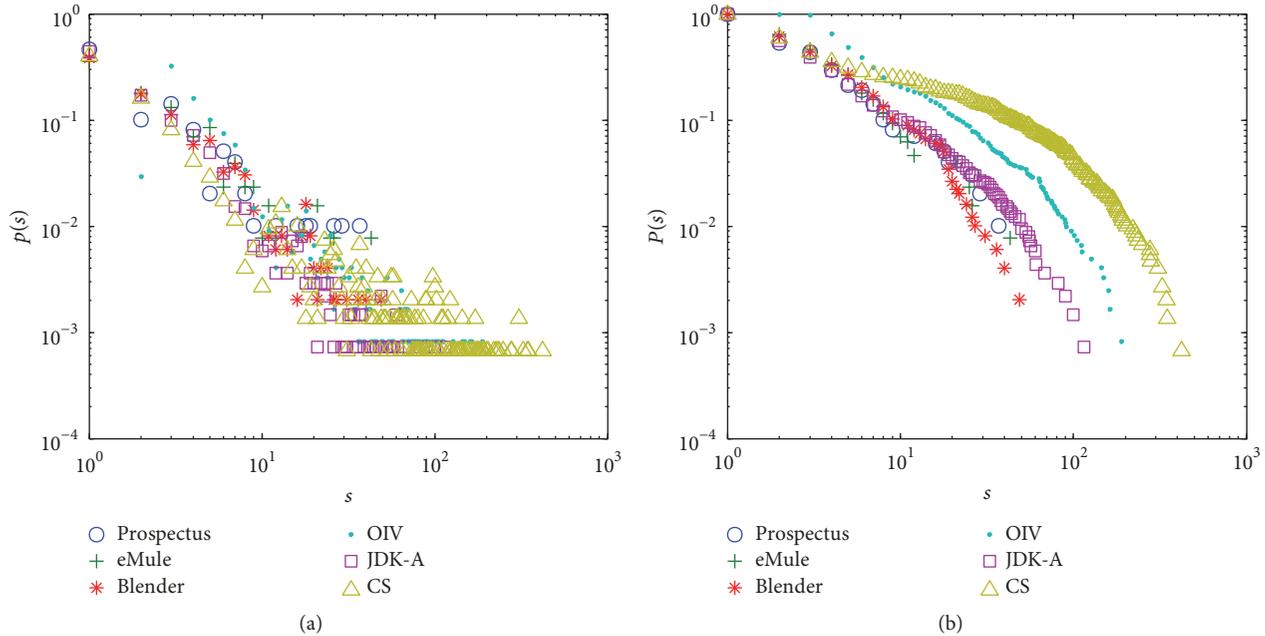


FIGURE 5: Strength distributions and cumulative strength distributions of the modeled software systems. (a) and (b) show the strength distributions and cumulative strength distributions, respectively.

6. Conclusions

In this paper, we propose the Weighted Multi-Local-World (WMLW) model, in which the weight of the link and the strength of the node are considered. Through analyzing the weight of links and strength of nodes, it is theoretically proven that the distribution of strengths yields to a power-law distribution, and simulations show the correctness of the theoretical result. In addition, the degree distribution also follows a power-law distribution. The modeling of class diagrams of real software systems is found to be a good application of the WMLW model. In our future work, we will find its more practical applications by improving the model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key Research Technology and Development Program of China (Grant no. 2016YFB0800404), the National Natural Science Foundation of China (Grants nos. 61702377, 61773175, 61572371, and 61403154), the China Postdoctoral Science Foundation

(Grant no. 2015M582272), and the Natural Science Foundation of Hubei Province of China (Grants nos. 2016CFB158 and 2017CFB426).

References

- [1] A. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [2] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [3] M. Newman, *Networks: An Introduction*, Oxford University Press, 2010.
- [4] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
- [5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. W. Hwang, "Complex networks: structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.
- [6] X. Li and G. Chen, "A local-world evolving network model," *Physica A: Statistical Mechanics and its Applications*, vol. 328, no. 1-2, pp. 274–286, 2003.
- [7] B. Wang, H. Tang, Z. Zhang, and Z. Xiu, "Evolving scale-free network model with tunable clustering," *International Journal of Modern Physics B*, vol. 19, no. 26, pp. 3951–3959, 2005.
- [8] Z. Pan, X. Li, and X. Wang, "Generalized local-world models for weighted networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 73, no. 5, Article ID 056109, 2006.
- [9] P. Li, Q. Zhao, and H. Wang, "A weighted local-world evolving network model based on the edge weights preferential selection," *International Journal of Modern Physics B*, vol. 27, no. 12, Article ID 1350039, 2013.
- [10] G. Chen, Z. Fan, and X. Li, "Modelling the complex internet topology," in *Complex Dynamics in Communication Networks*, L. Kocarev and G. Vattay, Eds., pp. 213–234, Springer Berlin Heidelberg, 2005.

- [11] Y. Qu, X. Guan, Q. Zheng et al., “Exploring community structure of software Call Graph and its applications in class cohesion measurement,” *The Journal of Systems and Software*, vol. 108, pp. 193–210, 2015.
- [12] G. Concas, M. Marchesi, C. Monni, M. Orrù, and R. Tonelli, “Software quality and community structure in java software networks,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 27, no. 7, pp. 1063–1096, 2017.
- [13] A. Decan, T. Mens, and P. Grosjean, “An empirical comparison of dependency network evolution in seven software packaging ecosystems,” *Empirical Software Engineering*, 2018.
- [14] R. Kikas, G. Gousios, M. Dumas, and D. Pfahl, “Structure and evolution of package dependency networks,” in *Proceedings of the 14th IEEE/ACM International Conference on Mining Software Repositories (MSR '17)*, pp. 102–112, May 2017.
- [15] C. Y. Chong and S. P. Lee, “Automatic clustering constraints derivation from object-oriented software using weighted complex network with graph theory analysis,” *The Journal of Systems and Software*, vol. 133, pp. 28–53, 2017.
- [16] M. Joblin, S. Apel, and W. Mauerer, “Evolutionary trends of developer coordination: a network approach,” *Empirical Software Engineering*, vol. 22, no. 4, pp. 2050–2094, 2017.
- [17] Y. Ding, B. Li, and P. He, “An improved approach to identifying key classes in weighted software network,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 3858637, 9 pages, 2016.
- [18] A. Barrat, M. Barthélemy, and A. Vespignani, “Weighted evolving networks: coupling topology and weight dynamics,” *Physical Review Letters*, vol. 92, no. 22, p. 228701, 2004.
- [19] B.-H. Wang, W.-X. Wang, and T. Zhou, “A weighted complex network model driven by traffic flow,” *Physics*, vol. 4, p. 012, 2006.
- [20] S. Valverde and R. V. Solé, “Hierarchical small worlds in software architecture,” *Disordered Systems and Neural Networks*, 2007, <https://arxiv.org/abs/cond-mat/0307278>.
- [21] R. Albert and A.-L. Barabási, “Topology of evolving networks: local events and universality,” *Physical Review Letters*, vol. 85, no. 24, pp. 5234–5237, 2000.
- [22] A. L. Barabási, R. Albert, and H. Jeong, “Mean-field theory for scale-free random networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 272, no. 1, pp. 173–187, 1999.

