A Generalized Configuration Model With Triadic Closure

Ruhui Zhang, Duan-Shin Lee, Senior Member, IEEE, and Cheng-Shang Chang, Fellow, IEEE

Abstract—In this paper we present a generalized configuration model with random triadic closure (GCTC). This model possesses five fundamental properties: large clustering coefficient, power law degree distribution, short path length, non-zero Pearson degree correlation, and existence of community structures. We analytically derive the Pearson degree correlation coefficient and the clustering coefficient of the proposed model. We select a few datasets of real-world networks. By simulation, we show that the GCTC model matches very well with the datasets in terms of Pearson degree correlations and clustering coefficients. We also test three well-known community detection algorithms on our model, the datasets and other three prevalent benchmark models. We show that the GCTC model performs equally well as the other three benchmark models. Finally, we perform influence diffusion on the GCTC model using the independent cascade model and the linear threshold model. We show that the influence spreads of the GCTC model are much closer to those of the datasets than the other benchmark models. This suggests that the GCTC model is a suitable tool to study network science problems where degree correlation or clustering plays an important role.

Index Terms—Clustering coefficient, community detection, configuration model, degree correlation, influence diffusion.

I. INTRODUCTION

NETWORK science is an inter-disciplinary field that formulates research problems arising in science and engineering into problems in graphs. To study successfully these diversified problems, researchers need suitable graph models. It would be nice to have a random graph model that possesses a rich set of topological properties and can be used to study a wide variety of problems in network science. In the mean time, it would be nice that the random graph model is simple enough to allow mathematical studies. There are five properties that are commonly observed in many real-life networks in science and engineering [1], [2]. They are

1) large clustering coefficient;
2) short average path length;
3) scale-free degree distribution, i.e. power law distribution;
4) assortative or disassortative degree correlation, and
5) existence of community structures.

In this paper we propose a random network model that possesses the five properties above. Many research problems in network science tie closely with these five properties. For instance, it is well known that diseases transmit efficiently among people in densely connected clusters, such as members in a household [3], [4]. It is also well known that clustering affects significantly the resilience of a network [5]. It is also commonly observed that densely connected clusters affect significantly how opinions spread in a network [6], [7]. To study these problems, one needs a graph model that is rich in transitivity. Authors in [8], [9], [10] showed that degree correlations and power law degree distributions have strong influence to the threshold of an epidemic. To study epidemic or influence diffusion problems, one may need a graph model that possesses degree correlations. The purpose of this paper is to propose a random graph model that is suitable to be used to study a broad range of problems. Our random model has all the five properties listed in the previous paragraph and is simple enough to allow mathematical analysis. It has a rich set of parameters and allows users to match its degree distribution, clustering coefficient, and degree correlation with those of a real-world dataset.

Our proposed model is based on configuration models originally proposed by Bender [11]. A configuration model is defined and constructed for a given degree sequence. It can have a power law degree distribution, if the given degree sequence is drawn from a power law distribution. It possesses a small world property. However, it lacks a community structure, and its clustering coefficient and degree correlation are asymptotically small as the network grows in size. Lee et al. [12] proposed a generalized configuration model, which adds a positive or a negative degree correlation to the configuration model. Lee et al. [12] achieved a non-zero degree correlation by partitioning stubs of edges into blocks and connecting stubs according to certain rules. In this paper, we propose to add another layer of blocks to emulate an artificial structure of communities. The result is a random graph model that possesses the fifth property listed in the first paragraph of this section. In addition, we propose to add triangles into the model to create a significant clustering coefficient. In literature, there are several proposals to achieve this objective. We refer the
reader to [13], [14] and the references therein. In this paper we propose to add triangles by performing triadic closure operations [15], [16]. In social science, triadic closure means that there is an increased likelihood that two people, who have a friend in common, will become friends [15]. Triadic closure has been observed in many real-world networks. For instance, Kossinets and Watts found clear evidence of triadic closure by taking multiple snapshots on an email communication network using a dataset consisting of 22,000 students at a large U.S. university [17]. Moreover, Leskovec et al. [18] analyzed the properties of triadic closure in online social networks of LinkedIn, Flickr, Del.icio.us, and Yahoo! Answers. In this paper we propose a random triadic closure operation. That is, all connected triangles that are not triangles yet are closed and become triangles with some probabilities. With these two new features, the new model possesses large transitivity and community structures. We call the new model generalized configuration model with triadic closure (GCTC). The detail construction algorithm of the GCTC model will be presented in Section II. We mention some research work in the literature that is related to triadic closure. Zhou et al. [19] proposed a network embedding method that takes the status of triads into consideration. Their method can learn representation vector for each vertex at different time points. Hofstad et al. [20] used term “triadic closure” to refer to triangles in a network. Hofstad et al. studied configuration models with power law degree distributions. Specifically, they showed that the local clustering coefficient of vertices with degree \( k \) and network size \( n \) is of the order \( n^{5-2\tau}k^{-2(3-\tau)} \) for \( \tau \in (2, 3) \), where \( \tau \) is the exponent of the power law degree distribution [20]. Clustering coefficients have been a widely accepted metric to measure triadic closure. Yin et al. [21] proposed new metrics to measure triadic closure in directed graphs.

We now outline the contributions of this paper. First, we derive closed form expressions for the Pearson degree correlation coefficient and the clustering coefficient of the new proposed GCTC model. We mention that there are other proposals of random networks that possess the five fundamental properties listed above. For example, Toivonen et al. [2] proposed a growth model that possesses all the five properties. It is not clear if the Pearson degree correlation coefficient of this model is mathematically tractable. Second, we examine whether the GCTC network is a suitable model to study various research problems in network science. We choose four datasets of networks collected in the real world. To make a comparison, we choose three random network models that are often used as benchmark models to evaluate community detection algorithms. The three models are stochastic blockmodel (SBM) [22], [23], Lancichinetti-Fortunato-Radicchi benchmark (LFR) model [24], [25] and artificial benchmark for community detection (ABCD) model [26]. Through simulations, we show that by choosing parameters properly the GCTC model can match better with the datasets in terms of the Pearson degree correlations and the clustering coefficients than the SBM, the LFR model and the ABCD model. Further, we compare the GCTC model with the SBM, the LFR model and the ABCD model using three well known community detection algorithms. We show that the GCTC model performs equally well as the other three benchmark models for community detection. Finally, we study influence diffusion in real-world networks, the GCTC model, the SBM, LFR and ABCD models. We study this problem using both an independent cascade (IC) model [27], [28] and a linear threshold (LT) model [27], [29]. We find that the fraction of influenced nodes observed in the GCTC model is much closer to that of the real-world datasets, compared with the fraction corresponding to the SBM model, LFR, and the ABCD models. This implies that the GCTC network is a more suitable model to study the influence diffusion problem. These results seem to indicate that degree correlation and clustering coefficient are less relevant to the community detection problem, but are crucial to the influence diffusion problem.

The organization of this paper is as follows. In Section II, we present a construction algorithm for the GCTC model. Then, we briefly review several basic analytical results of the generalized configuration model in Section III. In Section IV we analyze the Pearson degree correlation coefficient of the GCTC model. In Section V we analyze the clustering coefficient of the GCTC model. In Section VI we present numerical and simulation results. Finally, we present the conclusions of this paper in Section VII.

II. CONSTRUCTION ALGORITHM

In this section we present a construction algorithm for the GCTC model. Recall that in a standard configuration model, two ends of an edge are called “stubs”. To construct a standard configuration model, an unconnected stub is connected to another randomly selected stub among all unconnected stubs [30]. This construction algorithm creates a random network that has asymptotically vanishing Pearson degree correlation coefficient as the network becomes large. To introduce non-zero Pearson degree correlation, Lee et al. [12] partition stubs into blocks according to vertex degrees. To introduce positive (resp. negative) correlation, the selected permutation function associates blocks of large (resp. small) degrees with another block of large degrees. Stubs in block \( i \) are designated into type 1 stubs and type 2 stubs. An unconnected type 1 stub in block \( i \) is connected to a randomly selected unconnected type 1 stub in the associated block of block \( i \). An unconnected type 2 stub is connected to an unconnected type 2 stub randomly selected in all blocks. The construction algorithm of GCTC model is similar to that of a generalized configuration model, except that it has two additional features. First, GCTC model has two layers of blocks. Stubs are divided into macroscopic blocks, which model communities. In each macroscopic block, stubs are further divided into microscopic blocks, which are used to create a non-zero Pearson degree correlation. To achieve this, each stub is designated to one of three types. Type 1 and type 2 stubs provide intra-community connections and the non-zero degree correlation. Type 3 stubs provide inter-community as well as inter-community connections.

We describe the construction algorithm of GCTC model in details. There are \( c \) communities, where \( c \geq 1 \). Community \( i \),
where \( i = 1, 2, \ldots, c \), has \( n_i \) vertices. Let \( n \) be the total number of vertices in the network. It follows that

\[
n = \sum_{i=1}^{c} n_i.
\]

We note that the size of different communities can be distinct. That is, it is possible that \( n_i \neq n_j \) for some \( i \neq j \). Denote \( k_{ij} \) as the degree of the \( j \)-th vertex in community \( i \). Define \( m_i \) such that

\[
2m_i = \sum_{j=1}^{n_i} k_{ij}.
\]

The quantity \( 2m_i \) in (1) is the total number of stubs attached to vertices in community \( i \). The total number of edges in the network is \( m \), where

\[
m = \sum_{i=1}^{c} m_i.
\]

For each community \( i \), the edges are connected in the following ways. A degree \( k \) vertex in community \( i \) has \( k_i \) stubs. In this community \( i \), we arrange the stubs associated with the vertices in an ascending order of the vertex degrees. We then partition the stubs into \( b_i \) blocks evenly. That is, each block has the same number of stubs. Denote block \( b_{ij} \) as the block \( j \) in community \( i \). To create degree-degree correlations, we select a permutation function \( h_i(b_{ij}) \) for block \( b_{ij} \). Specifically, block \( b_{ij} \) is associated with block \( b_{ii} \), if \( h_i(b_{ij}) = b_{ii} \). In addition, \( h_i \) is selected such that \( h_i(h_i(b_{ij})) = b_{ij} \). We then classify the stubs in each block into three types proportionally. Denote the ratio of type 1 and type 2 stubs to the total stubs in each block in community \( i \) as \( r_i \in [0, 1] \) and the ratio of type 1 stubs to the type 1 and type 2 stubs in each block in community \( i \) as \( q_i \in [0, 1] \). Suppose \( r_i \) and \( q_i \) are given. For block \( b_{ij} \), randomly designate \( 2m_i q_i r_i / b_i \) stubs as type 1 stubs, and randomly designate \( 2m_i (1 - q_i) r_i / b_i \) stubs as type 2 stubs. Designate the rest stubs in block \( b_{ij} \) as type 3 stubs. To make a connection, one randomly picks an unconnected stub, say stub \( s \). If \( s \) is a type 1 stub in block \( b_{ij} \) of community \( i \), connect it with a randomly selected unconnected type 1 stub in block \( h_i(b_{ij}) \) and connect it to \( s \). If \( s \) is a type 2 stub, randomly select an unconnected type 2 stub in community \( i \) and connect it to \( s \). If \( s \) is a type 3 stub, randomly select an unconnected type 3 stub in the network and connect the stub to \( s \). These edges are referred to as regular edges.

Next, we apply triadic closure operations to increase the number of triangles in the network. The edges added into the network by the triadic closure operations are called transitive edges. We examine all pairs of unconnected vertices in the network. For each pair of unconnected vertices, say vertices \( A \) and \( B \), if \( A \) and \( B \) have \( d \) common neighbors, we connect \( A \) and \( B \) with probability \( a_d \). With probability \( 1 - a_d \), \( A \) and \( B \) remain unconnected. The construction algorithm for the GCTC model is shown in Algorithm 1.

---

**Algorithm 1: Construction Algorithm.**

**Inputs:** Degree sequence \( \{k_{ij}: i = 1, 2, \ldots, c, j = 1, 2, \ldots, n_i\} \), parameters \( \{b_i : i = 1, 2, \ldots, c\}, \{q_i : i = 1, 2, \ldots, c\}, \{r_i : i = 1, 2, \ldots, c\}, \{h_i : i = 1, 2, \ldots, c\} \) and \( \{a_d : d = 1, 2, \ldots, n - 2\} \).

**Outputs:** graph \( G = (V,E) \).

1: for \( i = 1, 2, \ldots, c \) do
2: For community \( i \), create \( 2m_i \) stubs from degree sequence \( \{k_{ij}: j = 1, 2, \ldots, n_i\} \) and arrange the stubs in an ascending order according to the degrees;
3: Divide \( 2m_i \) stubs into \( b_i \) blocks evenly;
4: for \( j = 1, 2, \ldots, b_i \) do
5: For block \( j \) of community \( i \), randomly designate \( 2m_i q_i r_i / b_i \) stubs as type 1 stubs, and randomly designate \( 2m_i (1 - q_i) r_i / b_i \) stubs as type 2 stubs;
6: Designate rest stubs in block \( j \) as type 3 stubs;
7: end for
8: end for
9: while there are unconnected stubs do
10: Randomly select a stub. Assume that the stub is in block \( j \) of community \( i \);
11: if type 1 stub then
12: connect this stub with a randomly selected type 1 unconnected stub in block \( h_i(j) \) in community \( i \);
13: else if type 2 stub then
14: connect this stub with a randomly selected type 2 unconnected stub in community \( i \);
15: else
16: connect this stub with a randomly selected type 3 stub among all type 3 stubs that are unconnected in the network;
17: end if
18: end while
19: for each unconnected pair of vertices do
20: if these two vertices have \( d \) common neighbors then
21: with probability \( a_d \), these two vertices are connected with a transitive edge, and with probability \( 1 - a_d \) leave these two vertices unconnected;
22: end if
23: end for

---

**III. Review of the Generalized Configuration Model.**

In this section we review some basic results of the generalized configuration model in [12]. These results will be used to derive Pearson degree correlation coefficient of the GCTC model in Section IV and the clustering coefficient of the GCTC model in Section V. We note that the generalized configuration model is a special case of the GCTC model where \( c = 1, r_1 = 1, b_1 = b_1, q_1 = q_1, h_1(i) = h_1(i) \) and \( a_0 = 0 \). We drop subscripts and use notation \( b, q, \) and \( h(i) \) to denote the number of blocks, the fraction of type 1 stubs, and the permutation function of block \( i \), respectively.

Let \( Z \) be the degree of a randomly selected vertex in the generalized configuration network. Then, the pmf of \( Z \) is \( \{p_k\} \) and

\[
E[Z] = \sum_{k=0}^{\infty} kp_k.
\]
The following assumption is crucial to the analysis. This assumption is stringent. We refer readers to [12] for additional information on this assumption.

**Assumption 1:** The degree distribution \( \{p_{k}\} \) is said to satisfy this assumption if one can find mutually disjoint sets \( H_1, H_2, \ldots, H_b \), such that

\[
\bigcup_{i=1}^{b} H_i = \{0, 1, 2, \ldots \}
\]

and

\[
\sum_{k \in H_i} kp_k = E[Z]/b
\]

for all \( i = 1, 2, \ldots, b \). In addition, we assume that the degree sequence \( k_1, k_2, \ldots, k_b \) sampled from the distribution \( \{p_{k}\} \) can be evenly placed in \( b \) blocks. That is, each block has the same number of stubs. In addition, stubs that belong to vertices of the same degrees are placed in the same block. Mathematically speaking, there exist mutually disjoint sets \( H_1, H_2, \ldots, H_b \) that satisfy

1. \( \bigcup_{i=1}^{b} H_i = \{1, 2, \ldots, n\} \)
2. \( k_i \neq k_j \) for any \( i \in H_{\ell_1}, j \in H_{\ell_2}, \ell_1 \neq \ell_2 \), and
3. \( \sum_{j \in H_i} k_j = 2 m/b \) for all \( i = 1, 2, \ldots, b \).

We randomly select a stub in the range \([1, 2m]\). Denote this stub by \( t \). Let \( v \) be the vertex, with which stub \( t \) is associated. Let \( Y \) be the degree of vertex \( v \). Since the stub is randomly selected and vertices with degree \( y \) have \( np_y \) stubs. We have

\[
\Pr(Y = y) = \frac{np_y}{2m} = \frac{y p_y}{E[Z]},
\]

Now connect stub \( t \) to a randomly selected stub. Let this stub be denoted by \( s \). Let \( u \) be the vertex, with which \( s \) is associated, and let \( X \) be the degree of vertex \( u \). We assume that Assumption 1 holds. Suppose \( x \) is a degree in set \( H_i \). The total number of stubs which are associated with vertices with degree \( x \) is \( nxp_x \). By Assumption 1, all \( nxp_x \) stubs are in block \( i \). There are two cases, in which stub \( t \) connects to stub \( s \). In the first case, stub \( t \) is of type 1. This occurs with probability \( q \). In this case, stub \( s \) must be a type 1 stub and belong to a vertex with a degree in block \( h(i) \). With probability

\[
\frac{q nxp_x}{2mq/b - \delta_{i,h(i)}},
\]

the construction algorithm in Section II connects \( t \) to \( s \). In (5) \( \delta_{i,j} \) is the Kronecker delta, is equal to one if \( i = j \), and is equal to zero otherwise. In the second case, stub \( t \) is of type 2. This occurs with probability \( 1 - q \). In this case, stub \( s \) can be associated with a degree in any block. With probability

\[
\frac{(1 - q) nxp_x}{2m(1 - q) - 1},
\]

the construction algorithm connects stub \( t \) to stub \( s \). Combining the two cases in (5) and (6), we have

\[
\Pr(X = x|Y = y) = \frac{q^2 nxp_x}{2mq/b - \delta_{i,h(i)}} + \frac{(1 - q)^2 nxp_x}{2m(1 - q) - 1}
\]

for \( y \in H_{h(i)} \). If \( y \in H_j \) for \( j \neq h(i) \),

\[
\Pr(X = x|Y = y) = \frac{(1 - q)^2 nxp_x}{2m(1 - q) - 1}.
\]

Now assume that the network is large. That is, we consider a sequence of constructed graphs, in which \( n \to \infty \), \( m \to \infty \), while keeping \( 2m/n = E[Z] \). Under this asymptotic, (7) and (8) converge to

\[
\Pr(X = x|Y = y) \to \begin{cases} 
\frac{q^2}{E[Z]}xp_x, & y \in H_{h(i)} \\
\frac{(1 - q)^2}{E[Z]}xp_x, & y \in H_j, j \neq h(i).
\end{cases}
\]

From (4) and (9) we obtain

\[
P(X = x, Y = y) = P(X = x|Y = y)P(Y = y)
\]

\[
= \begin{cases} 
\frac{q^2}{(E[Z])^2}xp_xp_y, & x \in H_i, y \in H_{h(i)} \\
\frac{(1 - q)^2}{(E[Z])^2}xp_xp_y, & x \in H_i, y \in H_j, j \neq h(i)
\end{cases}
\]

We next analyze the expected value of \( Y \) and the product \( XY \), respectively. From (4), we obtain

\[
E[Y] = \sum_{y} y \Pr(Y = y) = \sum_{j=1}^{b} \sum_{y \in H_j} y^2 p_y E[Z] = \frac{1}{E[Z]} \sum_{j=1}^{b} u_j,
\]

where

\[
u_j = \sum_{y \in H_j} y^2 p_y.
\]

From (10), we have

\[
E[XY] = \sum_{x} \sum_{y} xyP(X = x, Y = y)
\]

\[
= \frac{1 - q}{(E[Z])^2} \sum_{i=1}^{b} \sum_{j=1}^{b} u_i u_j + \frac{q^2}{(E[Z])^2} \sum_{i=1}^{b} u_i u_{h(i)}. \quad (11)
\]

We now consider \( E[Y|X] \). Denote the conditional expectation \( E[Y|X = x] \) by \( g(x) \). Assume that \( x \in H_i \) for some \( i \). From (9), we have

\[
g(x) = E[Y|X = x] = \sum_{i=1}^{b} \frac{(1 - q) u_i}{E[Z]} + \frac{q^2 u_{h(i)}}{E[Z]}. \quad (14)
\]

In addition, the analysis of the clustering coefficient needs the probability that two specific vertices are connected by a regular edge. Randomly select two vertices, say vertices \( A \) and \( B \). Denote the degrees of \( A \) and \( B \) by \( X_A \) and \( X_B \). Let the blocks of \( A \) and \( B \) be \( Q_A, Q_B \), respectively. We now consider the conditional connection probability
If \( h(i) \neq j \), vertices \( A \) and \( B \) can only be connected through a pair of type 2 stubs. A type 2 stub of vertex \( A \) connects to a type 2 stub of vertex \( B \) with probability \( (1 - q)k_B/2m(1 - q) \), since \( B \) has on average \( (1 - q)k_B \) type 2 stubs, there are totally \( 2m(1 - q) \) type 2 stubs in the network, and the connection is randomly selected. Since vertex \( A \) has \( (1 - q)k_A \) type 2 stubs on average, it follows that

\[
p_e(A, B) = \frac{(1 - q)^2k_Ak_B}{2m(1 - q)} = \frac{(1 - q)k_Ak_B}{2m}.
\]  

If \( h(i) = j \), vertices \( A \) and \( B \) can be connected through a pair of type 1 or type 2 stubs. In this case,

\[
p_e(A, B) = \frac{(1 - q)^2k_Ak_B + q^2k_Ak_B}{2m(1 - q)} = \frac{(1 - q + q)b_kk_Ak_B}{2m}.
\]  

IV. PEARSON DEGREE CORRELATION COEFFICIENT

In the section we analyze the Pearson degree correlation of the GCTC model. Recall that the GCTC model can have multiple communities. Stubs corresponding to vertices are first divided into communities. In each community, stubs are divided into blocks to create a non-zero Pearson degree correlation. This two level structure of stubs does not change the mathematical nature on how Pearson degree correlation being derived. However, it does increase the number of cases and the complexity of notations significantly. For this reason, we assume that there is one community in this section. In this special case, \( c = 1, r_1 = 1, b_i = b_j, q_i = q_j, h(i) = h(j) \) and \( a_d = a \). We drop subscripts and use notation \( b, q, h(i) \) and \( a \) to denote the number of blocks, the fraction of 1 stubs, the permutation function of block \( i \) and the probability that an unconnected pair of vertices are connected, respectively.

Pearson degree correlation is defined as the Pearson correlation coefficient of degrees at the two ends of a randomly selected edge. The GCTC model has two types of edges, regular edges and transitive edges. In this paper we analyze the correlation coefficient of degrees at the two ends of a randomly selected regular edge. Randomly select an edge among regular edges in the network. Let \( X \) and \( Y \) be the number of regular edges that the two ends of the edge have. Let \( X' \) and \( Y' \) be the number of transitive edges that the two ends of the edge have. We shall analyze the Pearson degree correlation

\[
\rho(X + X', Y + Y') \overset{\text{def}}{=} \frac{\text{Cov}(X + X', Y + Y')}{\sigma_{X+X'}\sigma_{Y+Y'}}
\]  

where \( \text{Cov}(X + X', Y + Y') \) is the co-variance of random variables \( X + X' \) and \( Y + Y' \) and \( \sigma_{X+X'} \) is the standard deviation of \( X + X' \).

Recall that in the generalized configuration model, the degree covariance is defined as \( \text{Cov}(X, Y) = E[XY] - E[X]E[Y] \) and \( \sigma_X = E[X^2] - (E[X])^2 \). In the GCTC model, the degree co-variance is defined as

\[
\text{Cov}(X + X', Y + Y') = E[XY] - E[X]E[Y] + E[XY'] - E[X]E[Y'] + E[XX'] - E[X]E[X'] + E[YY'] - E[Y]E[Y'].
\]  

And the product of standard deviations in the denominator of (15) is equal to

\[
\sigma_{X+X'}\sigma_{Y+Y'} = \sigma_X^2 + 2E[XX'] - E[X]E[X'] + E[Y]^2 - E[Y']^2.
\]  

Compared with \( \text{Cov}(X, Y) \), \( \text{Cov}(X + X', Y + Y') \) has additional expected terms related to \( X' \) and \( Y' \). Before we present the derivations of \( E[X] \), \( E[X'] \) and \( E[Y] \), we first show the following theorem. It is one of the main results in this paper. Due to space consideration, we refer the reader to [31] for the proof of this theorem.

**Theorem 2:** If \( h(i) = i \), then

\[
\text{Cov}(X + X', Y + Y') \geq \text{Cov}(X, Y) \geq 0.
\]  

We next analyze the terms needed to compute \( \text{Cov}(X + X', Y + Y') \) and \( \sigma_{X+X'} \). We observe that all the quantities of these expected terms are in the form of \( E[X^iY^j]/(E[X])^k \cdot (E[Y])^l \) for some integers \( i, j, k \) and \( l \). Therefore, we present the analysis of \( \text{Cov}(X + X', Y + Y') \) in the following three subsections. In Section IV-A, we show the analysis of \( E[X] \), \( E[X'] \) and \( E[Y] \). We next analyze the expected value \( E[X^iY^j]/(E[X])^k \cdot (E[Y])^l \) for integers \( i, j, k \) and \( l \) in Section IV-B. In Section IV-C, we present the final equations of \( \text{Cov}(X + X', Y + Y') \). Due to space limit, we refer the reader to [31] for the analysis of \( \sigma_{X+X'} \).

A. Analysis of \( E[X] \), \( E[X'] \), and \( E[Y] \)

To analyze \( E[X] \), \( E[X'] \), and \( E[Y] \), we shall first analyze the expected value of \( X' \). Recall that \( X \) and \( Y \) are the number of regular edges that two vertices at the two ends of a randomly selected regular edge. Let \( A \) and \( B \) denote the two vertices. \( X' \) and \( Y' \) are the number of transitive edges that \( A \) and \( B \) have, respectively. Number the \( X \) regular edges such that the first edge connects to \( B \). Along the \( i \)-th regular edge of \( A \) to reach the other side, where \( i = 2, 3, \ldots, X \), one finds \( Y_i \) regular edges. A graphical illustration is shown in Fig. 1.

Along the first edge, \( A \) has \( Y - 1 \) second neighbors and along the \( i \)-th edge, \( A \) has \( Y_i - 1 \) second neighbors. Totally, vertex \( A \) has

\[
Y - 1 + \sum_{i=2}^{X} (Y_i - 1) \tag{22}
\]

second neighbors. The total number of second neighbors in (22) can be overestimated, as some second neighbors can be counted
more than once. However, as the network size is large, the error is asymptotically small. We also remark that random variables $Y_i$, $i \leq 2$, are identically distributed for large networks. Their common distribution is the same as that of $Y$. The number of transitive edges that vertex $A$ has, given $X$ and $Y$, is

$$X' = \sum_{j=1}^{X-1} B_{1j} + \sum_{i=2}^{X} \sum_{j=1}^{Y-1} B_{ij},$$

(23)

where $\{B_{ij} : i \geq 1, j \geq 1\}$ is a doubly indexed sequence of independent and identically distributed Bernoulli random variables with success probability $a$. The conditional expectation of $X'$, given $X$ and $Y$, is

$$E[X'|X, Y] = E \left[ \sum_{j=1}^{X-1} B_{1j} + \sum_{i=2}^{X} \sum_{j=1}^{Y-1} B_{ij} \right| X, Y]$$

$$= a \cdot (Y - 1) + \sum_{i=2}^{X} a \cdot E[Y_i - 1|X].$$

(24)

Since $Y_i$ and $Y$ are identically distributed for all $i$, the preceding equation can be rewritten as

$$E[X'|X, Y] = a(Y - 1 + (X - 1)E[Y|X] - (X - 1)).$$

(25)

Taking expectation with respect to $X$ and $Y$, we have

$$E[X'] = aE[Y] - 1 + E[XY] - E[Y] - E[X] + 1$$

$$= aE[XY] - E[X].$$

(26)

Next, we analyze $E[XY']$. From (23) and similar to (24), we have

$$E[XY'|X, Y] = E \left[ \left( \sum_{j=1}^{X-1} B_{1j} + \sum_{i=2}^{X} \sum_{j=1}^{Y-1} B_{ij} \right) Y \right| X, Y]$$

$$= a((Y - 1)Y + (X - 1)Y \cdot E[Y - 1|X])$$

$$= a(Y^2 - XY + XYE[Y|X] - YE[Y|X]).$$

It follows that

$$E[XY] = a(E[Y]^2 - E[XY] + E[XYE[Y|X]]$$

$$- E[YE[Y|X]]).$$

(27)

We see that the terms on the right of the preceding expression are in the form of $E[X'Y']^2(E[Y'|X])^k(E[X'|Y])^l$ for some integers $i$, $j$, $k$ and $l$. We shall analyze these terms in the next subsection.

Finally, we analyze $E[XY']$. From (23), we have

$$E[XY'|X, Y] = a^2(Y - 1 + (X - 1)E[Y|X] - (X - 1))$$

$$\cdot (X - 1 + (Y - 1)E[X|Y] - (Y - 1)).$$

Then, we obtain

$$E[XY'] = a^2(-2E[X^2] + 2E[X^2Y]$$

$$- 2E[XYE[Y|X]] + 2E[YE[X|Y]]$$

$$+ E[YE[X|Y]] + E[X]E[Y] + E[Y]E[X] - 2E[Y]E[X|Y]$$

$$+ E[XYE[Y|X]]).$$

(28)

**B. Analysis of $E[XY']^2(E[y|X])^k(E[X'|Y])^l$**

We have observed that all quantities of expected values in (19) and (20) are in the form of $E[X'Y']^2(E[Y|X])^k(E[X'|Y])^l$ for some integers $i$, $j$, $k$ and $l$. Instead of tediously presenting the derivations of all expectation terms needed to compute the covariance and the variance, we choose a more complex term, that is $E[X^2|E[Y|X]]^2$, and derive it in this subsection. The derivation of other terms is similar, and is omitted. We refer the reader to [31] for the derivation.

With (14), we have

$$E[X^2(E[Y|X])^2] = \sum_x (xg(x))^2P(X = x)$$

$$= \sum_{i=1}^{b} \sum_{x \in H_i} \left( \sum_{i=1}^{b} \frac{(1 - q)u_i}{E[Z]} + \frac{q_{u_{(i)}}}{E[Z]} \right)^2$$

$$\cdot x^2 \frac{xp_x}{E[Z]}$$

$$= (1 - q)^2 \left( \frac{E[Z]^2}{E[Z]} \right)^2 \frac{E[Z^2]}{E[Z]^2}$$

$$+ 2(1 - q)q \frac{E[Z]}{E[Z]} \sum_{i=1}^{b} u_{(i)}t_i$$

$$+ q^2b^2 \frac{1}{E[Z]} \sum_{i=1}^{b} (u_{(i)}^2) t_i,$$
where
\[ t_i = \sum_{x \in H_i} x^i p_x. \] (29)

C. Analysis of \( \text{Cov}(X + X', Y + Y') \)

In this section we present an analysis of \( \text{Cov}(X + X', Y + Y') \). We first quote two results from Lee et al. [12]. Specifically,

\[ E[X] = \frac{E[Z^2]}{E[Z]} \] (30)

\[ E[XY] = \frac{1 - q}{(E[Z])^2} \sum_{i=1}^{n} \sum_{j=1}^{b} u_i u_j + \frac{q b}{(E[Z])^2} \sum_{i=1}^{b} u_i u_{h(i)} \] (31)

Substituting (30), (26), (31), (27), and (28) into (19), we obtain

\[ \text{Cov}(X + X', Y + Y') = \alpha_0 + \sum_{i=1}^{n} \beta_i W_i, \] (32)

where

\[ \alpha_0 = 2a \frac{a E[Z^2] - E[Z] + E[Z]}{(E[Z])^3} \left( E[Z] E[Z]^3 - (E[Z]^2)^2 \right) \]

\[ \beta_1 = \frac{q}{(E[Z])^2} + 2a \frac{q}{(E[Z])^2} \left( \frac{(1-q)E[Z]^2}{E[Z]} - 1 \right) \]

\[ + a^2 \left( (1-q)E[Z]^2 + qE[Z]^2 \right) - 2(2-q^2)E[Z^2]E[Z] \]

\[ \frac{(E[Z])^4}{(E[Z])^4} \]

\[ \beta_2 = 2a^2 - \frac{q}{(E[Z])^2} \]

\[ \beta_3 = -2a \frac{q^2}{(E[Z])^2} \left( 1 - \alpha + a \frac{1-qE[Z]}{E[Z]} \right) \]

\[ \beta_4 = 2a \frac{q^3 b}{(E[Z])^3} \left( 1 - \alpha - aq + a \frac{1-qE[Z]}{E[Z]} \right) \]

\[ \beta_5 = a^2 \frac{q^4 b^2}{(E[Z])^4} \]

\[ W_1 = b \sum_{i=1}^{b} u_i \sum_{j=1}^{b} u_j \] (33)

\[ W_2 = b \sum_{i=1}^{b} t_i \sum_{j=1}^{b} u_j \] (34)

\[ W_3 = b \sum_{i=1}^{b} u_i \sum_{j=1}^{b} t_j \] (35)

\[ W_4 = b \sum_{i=1}^{b} u_i \sum_{j=1}^{b} u_j \] (36)

\[ W_5 = b \sum_{i=1}^{b} u_i \sum_{j=1}^{b} u_j \] (37)

In (33) and (34), sequences \( \{u_i\} \) and \( \{t_i\} \) are defined in (12) and (29), respectively.

V. CLUSTERING COEFFICIENT

In this section we analyze the local clustering coefficient of the GCTC model. We remark that in this section we study the clustering coefficient of a special case in which there is only one community. We also remark that it is easy to extend the analysis to GCTC models with more than one community. We choose to present the result of the special case in order to keep notational simplicity.

Let \( A \) be a randomly selected vertex in a random network. Let \( k \) be the degree of \( A \). The local clustering coefficient of vertex \( A \) is defined as

\[ C_A(k) = \frac{\text{number of connected pairs of neighbors of } A}{\text{number of pairs of neighbors of } A} \]

\[ = \frac{\text{number of connected pairs of neighbors of } A}{k(k-1)/2}. \] (38)

If \( k \leq 1 \), \( C_A(k) \) is defined to be zero. We distinguish between regular edges and transitive edges. Assume that vertex \( A \) has \( k \) regular edges and \( k' \) transitive edges. We denote the local clustering coefficient of \( A \) by \( C_A(k, k') \). Let the \( k \) vertices connected with \( A \) by regular edges be denoted by \( U_1, U_2, \ldots, U_k \). Let the \( k' \) vertices connected with \( A \) by transitive edges be denoted by \( V_1, V_2, \ldots, V_{k'} \).

We analyze the numerator of (38). Obviously, if \( k = 0 \), \( C_A(0, k') = 0 \). For general \( k \geq 1 \) and \( k' \geq 0 \), we claim that

\[ C_A(k, k') = \frac{k}{2} \frac{a + k' + \frac{k'}{2}}{k} \times \frac{a}{k}, \] (39)

with the convention that

\[ (i, j) = 0 \text{ if } i < j. \]

To analyze (39) we consider six types of triangles as shown in Fig. 2. We consider type 1 triangles shown in panel (a) of Fig. 2. The expected number of type 1 triangles is

\[ \sum_{k_1, k_2} \binom{k}{2} p_{U_1} p_{U_2} p_{k_1} p_{k_2}, \]

where \( k_1 \) and \( k_2 \) are the degrees of vertices \( U_1 \) and \( U_2 \), respectively. Since \( 2m = n E[Z] \), it follows that the expected number of type 1 triangles in the last expression approaches to zero as the network size \( n \) is large. Note that type 5 triangles in panel (e) of Fig. 2 also require vertices \( U_1 \) and \( U_2 \) be connected by regular edges. By the same argument, it is easy to see that the expected number of type 5 triangles also goes to zero as the network gets large.

Now we consider the second type of triangles shown in panel (b) of Fig. 2. Vertices \( U_1 \) and \( U_2 \) are connected by a transitive edge. This transitive edge is formed because vertex \( U_2 \) is an unconnected second neighbor of \( U_1 \) through vertex \( A \). Thus, the expected number of type 2 triangles is
\[ \sum_{k_1, k_2} \binom{k}{2} (1-p_c(U_1, U_2) p_{k_1} p_{k_2}) \cdot a = \binom{k}{2} \cdot a. \]

This is the first term in the numerator of (39).

We next analyze type 3 triangles shown in panel (c) of Fig. 2. Note that transitive edge $AV_1$ can be formed in two types of event. The first type of event is the successful event of random triadic closure of the connected triples of $A$, $V_1$ and $U_1$. The second type of event is the successful event of random triadic closure of the connected triples of $A$, $V_1$ and one of the first neighbors of $A$ in the set $\{U_1, U_2, \ldots, U_k\}$ except $U_1$. If the transitive edge $AV_1$ in type 3 triangle is formed from the first type of event, the number of type 3 is $k'$. If the transitive edge $AV_1$ is formed from the second type of event, to form a type 3 triangle, $U_1$ and $V_1$ need to be connected by a regular edge. From the analysis of type 1 triangles, we know that the connected probability of $U_1$ and $V_1$ approaches to zero as the network size $n$ is large. To sum, the expected number of type 3 triangles is $k'$, which is the second term in the numerator of (39).

Now we consider type 4 triangles shown in panel (d) of Fig. 2. In order to form a transitive edge between $U_1$ and $V_1$, these two vertices must have at least one common neighbor by regular edges. Besides vertices $A$, $U_1$ and $V_1$, there are $n-3$ vertices in the network. Let $E_{n-3}$ be the event that there is at least one vertex in $n-3$ vertices that connects to both $U_1$ and $V_1$. Then, the expected number of type 4 triangles is

\[ kk' a P(E_{n-3}) = kk' a (1 - P(E_{n-3}^c)), \]

where $E_{n-3}^c$ is the complement of event $E_{n-3}$. Denote the $n-3$ vertices by vertices $1, 2, \ldots, n-3$.

\[ P(E_{n-3}^c) = \sum_{u,v,k_1,k_2,\ldots,k_{n-3}} \prod_{j=1}^{n-3} (1-p_c(j, U_1)(1-p_c(j, V_1)) \]

\[ \times p_{u} p_{v} p_{k_1} p_{k_2} \ldots p_{k_{n-3}}, \]

where $u$ and $v$ are the degrees of $U_1$ and $V_1$, and $k_j$ is the degree of vertex $j$ for $j = 1, 2, \ldots, n-3$. To evaluate $P(E_{n-3}^c)$, we substitute (16) or (17) into (41). For example,

denote $b_{v_1}$, $b_{v_2}$ and $b_j$ as the block index of $U_1$, $V_1$ and $j$, respectively. Suppose $b_{v_1} \neq h(b_j)$ and $b_{v_1} \neq h(b_j)$, we substitute (16) into (41) and have

\[ P(E_{n-3}) = 1 - \sum_{u,v,k_1,k_2,\ldots,k_{n-3}} \prod_{j=1}^{n-3} \frac{1 - (1-q) u k_j (1-q) v (k_j - 1)}{2 m} \]

\[ \times p_u p_v p_{k_1} p_{k_2} \ldots p_{k_{n-3}} \]

\[ = 1 - \sum_{u,v} \left( 1 - \frac{(1-q)^2 u v (E[Z]^2 - E[Z])}{(nE[Z])^2} \right)^{n-3} p_u p_v \]

\[ \leq 1 - \left( 1 - \frac{(1-q)^2 (E[Z]^2 - E[Z])}{(nE[Z])^2} \right)^{n-3} \]

\[ \to 0 \text{ as } n \to \infty, \quad (42) \]

where inequality (42) is due to Jensen’s inequality [32]. It follows that the expected number of type 4 triangles is zero in large networks. Note that substitution of (17) into (41) leads to the same result, i.e. the expected number of type 4 triangles is zero in large networks.

Finally, we consider the expected number of type 6 triangles shown in panel (f) of Fig. 2. Note that to form transitive edges $AV_1$ and $AV_2$, vertices $V_1$ and $V_2$ must be unconnected second neighbors of $A$ through some first neighbors $U_1$ and $U_2$. In panel (a), the two first neighbors of $A$ are distinct. In panel (b), vertices $V_1$ and $V_2$ have a common first neighbor of $A$.

Fig. 3. Type 6 triangles. Vertices $V_1$ and $V_2$ must be unconnected second neighbors of $A$ through some first neighbors $U_1$ and $U_2$. In panel (a), the two first neighbors of $A$ are distinct. In panel (b), vertices $V_1$ and $V_2$ have a common first neighbor of $A$.

By the same argument in (42), it is easy to show that the quantity above goes to zero as $n$ goes to infinity. Now we consider
the second case shown panel (b) of Fig. 3. Vertices \( V_1 \) and \( V_2 \) share a common first neighbor \( U_i \) with \( A \). The probability that the random selections of \( V_1 \) and \( V_2 \) are the same is \( \frac{1}{k} \). Thus, the expected number of type 6 triangles is

\[
\binom{k'}{2} \cdot \frac{a}{k^2}.
\]

This is the third term in the numerator on the right side of (39).

VI. SIMULATION RESULTS

In this section we present our numerical and simulation results. In Section VI-A we model real-world networks by GCTC networks. To make a comparison we also model the same real-world networks using SBM, LFR and ABCD models. In Section VI-B, we study whether the GCTC model is suitable to be a benchmark model for community detection algorithms. Finally, in Section VI-C we simulate influence diffusion in GCTC networks and compare the result with that in a real-world network.

A. Modeling Real-World Networks

In this section we would like to study if it is possible to choose a proper set of parameters such that the degree distribution, Pearson degree correlation coefficient and the clustering coefficient of the GCTC model match reasonably well with those of a real-life network. To study this problem, we choose four networks collected in the real life. Since we will study community detection algorithms on real-world networks in the next section, we choose real-world networks with a known community structure. Specifically, the four real-world networks that we choose are the Amazon network, the Email network, the Ogbn-arxiv network and the DBLP network. The Ogbn-arxiv data can be obtained from the OGB website. The other three network data are available at the Stanford website.

These four datasets need preprocessing before they can be used. The Amazon network has multiple overlapped communities. We keep only one community and delete all other vertices. As a result, it has only one community. For the Email network, we keep communities with more than fifty vertices, and delete smaller communities. For the Ogbn-arxiv network, we only keep two largest communities whose sizes are smaller than ten thousand vertices. For the DBLP network, we keep the largest three communities.

We construct GCTC graphs with Algorithm 1 using properly selected parameters to match the performance of the four real-world graphs. The Pearson degree correlation coefficient, the clustering coefficient, the average length of shortest path between randomly selected vertices, among others, are shown in Table I. Note that due to triadic closure operations, the total
number of edges \( m \) and the expected degree \( \mathbb{E}[Z] \) are not identically equal to those of the real-world networks. From column 7 and column 8, we see that the Pearson degree correlation coefficient and the clustering coefficient of the GCTC model match quite well with those of the four real-life networks.

To make a comparison, we also simulate three prevalent benchmark models that are often used to evaluate community detection algorithms. They are the SBM, the LFR model and the ABCD model. We briefly review these three models. The simplest SBM is a multi-graph with a given number of communities. Each vertex is assigned to a community. Undirected edges are placed independently between vertex pairs with probabilities that are only a function of the community membership of the vertices [22]. As a result, each vertex in an SBM has a Poisson distribution for its degree. To fit an SBM with an empirical network collected in the real world, one typically formulates a maximum likelihood problem to determine the parameters in the Poisson distributions. Unfortunately, the traditional SBM does not work well in the sense that it can not fit well with a wide range of network data. Karrer et al. [23] proposed a degree-corrected version of the SBMs. In the degree-corrected version, each vertex is associated with a new parameter, with which the parameters of Poisson distributions are multiplied and thus corrected. A maximum likelihood problem solves both the new parameters as well as the parameters of the Poisson distributions. We refer the reader to [23] for more details. In this paper, we choose the degree-corrected SBMs to fit with the four real-world networks. Next, we briefly review LFR model. An LFR graph is constructed first by sampling a power law distribution for a sequence of \( n \) degrees. Then a sequence of community sizes is sampled from another power law distribution. Clearly, the sum of all community sizes must be equal to the total number of vertices. Randomly assign each vertex to one community. Users of an LFR model also need to determine another parameter, the fraction of edges that connect two vertices in two distinct communities. Let \( \mu \) denote this parameter. Each vertex is randomly connected by a fraction \( 1 - \mu \) of its links with vertices within its community and a fraction \( \mu \) of its links with the other vertices of the network. Finally, we review ABCD model. The acronym ABCD stands for Artificial Benchmark for Community Detection. It is a modification of the LFR model in an attempt to make it be constructed faster. An ABCD model also has a parameter \( \xi \) called a mixing parameter. An ABCD model connects a fraction \( 1 - \xi \) of its edges to other vertices in the same community, and connects a fraction \( \xi \) of its edges to vertices globally including the vertices in the same community. Finally, we mention that the construction algorithm for LFR networks has a complexity of \( O(n^2) \). The construction algorithm for GCTC networks has a complexity of \( O(n^3) \).

To model a given real-world network, we measure the fraction of edges connecting two vertices in two distinct communities. We call this quantity the mixing parameter for the real-world network. For simplicity, we set \( \mu \) to this quantity and use it to construct LFR networks. We also use this quantity to calculate the value of \( \xi \) according to [26]. We then use \( \xi \) to construct ABCD networks. In Table I, we present the values of the mixing parameter for the real-life networks and simulated graphs. Moreover, from the entries in columns 7 and 8 in Table I, we see that the Pearson degree correlation coefficient and the clustering coefficient of the GCTC model match consistently better with those of the real-world networks than the SBM, the LFR model, and the ABCD model.

B. Community Detection

One of the possible applications of the GCTC model is to serve as a benchmark model for community detection algorithms. In this section we compare the performance of the GCTC model with three well known benchmark models.

We test the performance of the GCTC model as a benchmark model using three well known community detection algorithms. They are the walktrap algorithm [33], the leading eigenvector algorithm [34] and the fast greedy algorithm [35]. We use normalized mutual information (NMI) to measure the performance of a benchmark model. We briefly state the definition of NMI here and refer the reader to [23], [36] for more details. Let \( n_{ij} \) be the number of vertices in community \( i \) in the inferred community detection and in community \( j \) in the ground truth. Define joint probability \( P_r(i_1 = i, C_2 = j) = n_{ij}/n \) that a randomly selected vertex is in \( i \) in the inferred detection and \( j \) in the ground truth. Using this joint probability over the random variables \( C_1 \) and \( C_2 \), the NMI is defined as

\[
NMI(C_1, C_2) = \frac{2MI(C_1, C_2)}{H(C_1) + H(C_2)},
\]

where \( MI(C_1, C_2) \) is the mutual information between \( C_1 \) and \( C_2 \), \( H(C_1) \) is the entropy of random variable \( C_1 \), and \( H(C_2) \) is the entropy of random variable \( C_2 \). Higher values of NMI indicate a higher degree of consistency between the detected structure of communities and the ground truth. Since the Amazon network has only one community, we have not applied community detection algorithms on it. For each benchmark model, we simulate and generate one thousand graphs. For each graph, we apply the three community detection algorithms and compute the NMI values. We present the average of the NMI values in columns 10, 11 and 12 in Table I. From our simulation results, we see that the GCTC model performs better than the SBM, the LFR model, and the ABCD benchmark model when they model the Ogbn-arxiv network. For the other two real-world networks with leading eigenvector algorithm, the performance of the GCTC model is not the best. In fact, the performance of the GCTC model was the worst among the four benchmark models when the four models synthesize the Email network to evaluate the leading eigenvector algorithm. Since the NMI values of the GCTC model generally agree with those of the real-world networks and the other three random network models, we conclude that the GCTC model can serve well as a benchmark model for the evaluation of community detection algorithms. In addition, since the GCTC model performs the best in matching its \( \rho \) and \( C \) with those of the real-world networks, this study seems to imply that community detection problem is less sensitive to degree correlation and transitivity of a network. We observe that the mixing parameter for the DBLP network is very close to 1/2. This fact might make the DBLP network very difficult for all community detection
algorithms. Indeed, the NMI values of all the three community detection algorithms are very small.

C. Influence Diffusion

We simulate influence cascade in the four real-world networks, the GCTC, the SBM, the LFR and the ABCD networks. We present results in this section.

We simulate two most prevalent influence cascade models, the IC model and the LT model [28]. In both models, the state of a vertex can be either active or inactive at any time step. Initially at time zero, a certain number of vertices are selected to be active. They are called the seeds of the diffusion. In the independent cascade model, when an inactive vertex becomes activated, it will independently activate each of its currently inactive neighbors with probability $p$ in the next time step. Each active vertex has exactly one opportunity to influence its currently inactive neighbors. In our experiment, we set $p = 0.15$. In the linear threshold model, each inactive vertex, say $v$, computes the fraction of its active neighbors to its degree. The vertex switches to the active state if the fraction exceeds a threshold $t$. In the experiment we set $t = 0.35$. In both IC and LT models, the influence diffusion process unfolds in discrete time steps until no more vertices can be activated. We call the fraction of active vertices the influence spread of the diffusion process. This quantity is shown in columns IC and LT. From the entries in these two columns, we see that the GCTC model outperforms the other three models in predicting the influence spreads in both the IC model and the LT model in most cases. The only except is the Email network. In both the independent cascade model and the linear threshold model, the influence spreads of the four random network models are very close. GCTC is not the worst among the four models. Studies in [6], [7] show that densely connected clusters affect significantly how opinions diffuse in a network. Recall that the degree correlations and transitivity of the GCTC model have the best match with those of real-world networks. It seems to imply that the influence diffusion problem is quite sensitive to the degree correlation and the clustering coefficient of the network in which the influence diffuses. In Figs. 4 and 5 we present the influence spread in the DBLP network and the four random network models as functions of $p$ and $t$, respectively. From these two figures, we see that the influence spread of the real network can be very different from those in random networks for certain range of $p$ and $t$. However, the curves corresponding to the GCTC model trace closely with those of the real network.

VII. CONCLUSION

In this paper, we have presented a generalized configuration model with triadic closure. This model is an extension of the generalized configuration model with an additional layer of blocks and random triadic closure operations. The GCTC model possesses five most important properties of graphs that arise in network science. We have analyzed the Pearson degree correlation and clustering coefficient of the GCTC model. We have applied the GCTC model, the SBM, the LFR model and the ABCD model to model four networks collected in the real world. We have shown that by choosing parameters properly the GCTC model matches much better its clustering coefficient and Pearson degree correlation coefficient with those of the datasets than the other three random models. We have used the GCTC model, the SBM, the LFR model, the ABCD model and the four real-world networks to examine four community detection algorithms and the influence diffusion problem. We found that the GCTC model, the SBM, the LFR model, the ABCD model and the four real-world networks perform nearly equally well as a benchmark model for community detection with GCTC model slightly better than the other three models. We have simulated influence diffusion in the four models and the four real-world networks using the independent cascade model and the linear threshold model. We found that the fraction of vertices influenced in the GCTC model matches much better with that of real-world networks than the other three models.

REFERENCES

[1] M. Newman, Networks: An Introduction. New York, NY, USA: Oxford Univ. Press, 2010.
[2] R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski, “A model for social networks,” Physica A: Stat. Mechanics Appl., vol. 371, no. 2, pp. 851–860, 2006.
[3] E. Coupéchoux and M. Lelarge, “How clustering affects epidemics in random networks,” Adv. Appl. Probability, vol. 46, no. 4, pp. 985–1008, 2014.
[4] P. Trapman, “On analytical approaches to epidemics on networks,” Theor. Popul. Biol., vol. 71, no. 2, pp. 160–173, 2007.
[5] J. P. Gleeson, S. Melnik, and A. Hackett, “How clustering affects the bond percolation threshold in complex networks,” Phys. Rev. E, vol. 81, no. 6, 2010, Art. no. 066114.
[6] S. Morris, “Contagion,” Rev. Econ. Stud., vol. 67, no. 1, pp. 57–78, 2000.
[7] D. J. Watts and P. S. Dodds, “Networks, influence, and public opinion formation,” J. Consum. Res., vol. 34, pp. 441–458, 2007.

[8] M. Boguñá, R. Pastor-Satorras, and A. Vespignani, “Absence of epidemic threshold in scale-free networks with degree correlations,” Phys. Rev. Lett., vol. 90, 2003, Art. no. 028701.

[9] M. Boguñá and R. Pastor-Satorras, “Epidemic spreading in correlated complex networks,” Phys. Rev. E, vol. 66, 2002, Art. no. 047104.

[10] V. M. Eguíluz and K. Klemm, “Epidemic threshold in structured scale-free networks,” Phys. Rev. Lett., vol. 89, no. 10, 2002, Art. no. 108701.

[11] E. A. Bender and E. R. Canfield, “The asymptotic number of labelled graphs with given degree sequences,” J. Combinatorial Theory Ser. A, vol. 24, pp. 296–307, 1978.

[12] D.-S. Lee, C.-S. Chang, M. Zhu, and H.-C. Li, “A generalized configuration model with degree correlations and its percolation analysis,” Appl. Netw. Sci., vol. 4, no. 1, Dec. 2019, Art. no. 124.

[13] M. Newman, “Random graphs with clustering,” Phys. Rev. Lett., vol. 103, Jul. 2009, Art. no. 058701.

[14] C. Wang, O. Lizardo, and D. Hachen, “Algorithms for generating large-scale clustered random graphs,” Newt. Sci., vol. 2, pp. 403–415, Jul. 2014.

[15] A. Rapoport, “Spread of information through a population with sociostructural bias I: Assumption of transitivity,” Bull. Math. Biophysics., vol. 15, no. 4, pp. 523–533, Dec. 1953.

[16] D. Easley and J. Kleinberg, Networks, Crowds and Markets Reasoning About a Highly Connected World. Cambridge, U.K.:Cambridge Univ. Press, 2010.

[17] G. Kossinets and D. J. Watts, “Empirical analysis of an evolving social network,” Science, vol. 311, no. 5757, pp. 88–90, 2006.

[18] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, “Microscopic evolution of social networks,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2008, pp. 462–470.

[19] L. Zhou, Y. Yang, X. Ren, F. Wu, and Y. Zhuang, “Dynamic network embedding by modeling triadic closure process,” Proc. 32nd AAAI Conf. Artif. Intell., 30th Innov. Appl. Artif. Intell. Conf. 8th AAAI Symp. Educ. Adv. Artif. Intell., 2018, pp. 571–578.

[20] R. van der Hofstad, J. S. H. van Leeuwaarden, and C. Stegehuis, “Triadic closure in configuration models with unbounded degree fluctuations,” J. Stat. Phys., vol. 173, no. 3/4, pp. 746–774, 2018.

[21] H. Yin, A. R. Benson, and J. Ugander, “Measuring directed triadic closure with closure coefficients,” Newt. Sci., vol. 8, no. 4, pp. 551–573, Jun. 2020.

[22] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic block models: First steps,” Social Netw., vol. 5, no. 2, pp. 109–137, 1983.

[23] B. Karrer and M. E. J. Newman, “Stochastic blockmodels and community structure in networks,” Phys. Rev. E, vol. 83, Jan. 2011, Art no. 016107. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.83.016107

[24] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” Phys. Rev. E, vol. 78, no. 4, 2008, Art. no. 046110.

[25] A. Lancichinetti and S. Fortunato, “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities,” Phys. Rev. E, vol. 80, no. 1, 2009, Art. no. 016118.

[26] B. Kaminski, P. Prałat, and F. Theberge, “Artificial benchmark for community detection (ABCD)–fast random graph model with community structure,” Newt. Sci., vol. 9, no. 2, pp. 153–178, 2021.

[27] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network,” in Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2003, pp. 137–146.

[28] P. Shakarian, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo, “The independent cascade and linear threshold models,” in Diffusion in Social Networks. Berlin, Germany:Springer, 2015, pp. 35–48.

[29] A. Borodin, Y. Filmus, and J. Oren, “Threshold models for competitive influence in social networks,” in Proc. Int. Workshop Internet Network. Econ., 2010, pp. 539–550.

[30] Bélá Bollobás, “A probabilistic proof of an asymptotic formula for the number of labelled regular graphs,” Eur. J. Combinatorics, vol. 1, pp. 311–316, 1980.

[31] R. Chang, D.-S. Lee, and C.-S. Chang, “A generalized configuration model with triadic closure,” 2021, arXiv:2105.11688.

[32] W. Feller, An Introduction to Probability Theory and Its Application, vol. 1 and 2. New York, NY, USA: Wiley, 1971.

[33] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” in Proc. Int. Symp. Comput. Inf. Sci.2005, pp. 284–293.

[34] M. Newman, “Finding community structure in networks using the eigenvectors of matrices,” Phys. Rev. E, vol. 74, no. 3, 2006, Art. no. 036104.

[35] A. Clauset, M. E. Newman, and C. Moore, “Finding community structure in very large networks,” Phys. Rev. E, vol. 70, no. 6, 2004, Art. no. 066111.

[36] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, “Comparing community structure identification.” J. Stat. Mechanics: Theory Exp., vol. 2005, 2005, Art. no. P09008.

Ruhui Zhang received the B.S. degree in advertisement from Xiamen University, Xiamen, China, in 2014. She is currently working toward the Ph.D. degree with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Her research interests include network science, social networks, and economic networks.

Dah-Shin Lee (Senior Member, IEEE) received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 1987 and 1990. From 1990 to 1998, he was a Research Staff Member with the C&C Research Laboratory of NEC USA Inc., Princeton, NJ, USA. In 1998, he joined the Department of Computer Science of National Tsing Hua University. Since August 2003, he has been a Professor. He was the recipient of the Best Paper Award from the Y.Z. Hsu Foundation in 2006. He was the Editor of the Journal of Information Science and Engineering between 2013 and 2015. He was also an Associate Editor for Performance Evaluation between 2018 and 2021. His research interests include network science, game theory, machine learning and high-speed networks.

Cheng-Shang Chang (Fellow, IEEE) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 1986 and 1989, respectively. From 1989 to 1993, he was a Research Staff Member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Since 1993, he has been with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, where he is currently a Tsing Hua Distinguished Chair Professor. He is also the author of the book Performance Guarantees in Communication Networks (Springer, 2000) and the coauthor of the book Principles, Architectures and Mathematical Theory of High Performance Packet Switches (Ministry of Education, R.O.C., 2006). His research interests include network science, Big Data analytics, mathematical modeling of the Internet, and high-speed switching. Dr. Chang was the Editor of Operations Research from 1992 to 1999, IEEE/ACM TRANSACTIONS ON NETWORKING from 2007 to 2009, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING from 2014 to 2017. He is also the Editor-at-Large of the IEEE/ACM TRANSACTIONS ON NETWORKING. He is the Member of IFIP Working Group 7.3. He was the recipient of an Outstanding Innovation Award in 1992, IBM Faculty Partnership Award in 2001, Outstanding Research awards from the National Science Council, Taiwan, in 1998, 2000, and 2002, respectively, Outstanding Teaching awards from both the College of EECS and the University itself in 2003, the Merit NSF Research Fellow Award from the National Science Council, R.O.C. in 2011, Academic Award in 2011, National Chair Professorship in 2017 from the Ministry of Education, R.O.C., and 2017 IEEE INFOCOM Achievement Award. He was appointed as the first Y. Z. Hsu Scientific Chair Professor in 2002.