doi:10.1017/nws.2014.7

Algorithms for generating large-scale clustered random graphs

CHENG WANG

Department of Population Health and Disease Prevention, University of California, Irvine, A.I.R.Bldg 653,
Suite 2040H, 653 East Peltason Drive, Irvine, CA 92697, USA
(e-mail: wang.cheng@uci.edu)

OMAR LIZARDO and DAVID HACHEN

Department of Sociology, University of Notre Dame, 735 Flanner, Notre Dame, IN 46545, USA
(e-mail: {olizardo, dhachen}@nd.edu)

Abstract

Real-world networks are often compared to random graphs to assess whether their topological structure could be a result of random processes. However, a simple random graph in large scale often lacks social structure beyond the dyadic level. As a result we need to generate clustered random graph to compare the local structure at higher network levels. In this paper a generalized version of Gleeson's algorithm $G(V_S, V_T, E_S, E_T, S, T)$ is advanced to generate a clustered random graph in large-scale which persists the number of vertices $|V|$, the number of edges $|E|$, and the global clustering coefficient C_Δ as in the real network and it works successfully for nine large-scale networks. Our new algorithm also has advantages in randomness evaluation and computation efficiency when compared with the existing algorithms.

Keywords: *large-scale network, clustered random graph, generating algorithm*

1 Introduction

Random graphs are widely used to compare with real networks. A random graph preserves the number of vertices $|V|$ and the number of edges $|E|$ of the real network. This does work for small network with hundreds of or thousands of vertices and thousands or tens of thousands of edges. However, as the network size grows larger and larger, the simple random graph fails to reproduce the local structure beyond dyadic level which is correlated with non-zero clustering coefficient, "small world" phenomenon, and other important network characteristics.

There are at least four existing algorithms advanced to generate a random graph with clustering. However, none of these algorithms has been tested for large-scale networks. In this paper we go over these algorithms, examine their feasibility, advantages, and disadvantages, and make some revisions if necessary for generating clustered random graphs in large scale.

2 From simple random graph to clustered random graph

Networks in our real world usually share three common characteristics: i) Skewed degree distribution – most vertices have low nodal degrees but a small number, known as “hubs”, have high degrees (see Barabási & Albert, 1999; Newman, 2003); ii) “Small world” or “six degree of separation” phenomenon – the geodesic distance between most, if not all, pairs of vertices is limited (Travers & Milgram, 1969; de Sola Pool & Kochen, 1978/1979; Watts & Strogatz, 1998); and iii) Non-zero clustering coefficient – vertices in networks tend to stay in triangles[1] (see Simmel, 1908/1950; Heider, 1946, Cartwright & Harary, 1956; Davis, 1967, 1979; Granovetter, 1973; Krackhardt, 1998; Krackhardt & Handcock, 2006; Opsahl & Panzarasa, 2009)

Simple random graphs have long been used to compare with real networks. They are generated by adding edges between a set of n vertices at random. The first simple random graph was proposed by Erdős and Rényi (1959), denoted as $G(n, p)$, which has n nodes (identical to the number of vertices $|V|$) and each edge follows an independent formation probability $p \in (0, 1)$ (identical to the network density $\Delta = \frac{|E|}{|V|(|V|-1)/2} = \frac{2|E|}{|V|(|V|-1)} = \frac{2\frac{|E|}{|V|}}{|V|-1} = p$). Later Molloy and Reed (1995) developed a configuration model with a fixed degree sequence. However, by preserving the number of vertices $|V|$ and the number of edges $|E|$, the randomly wired network only successfully reproduces the network characteristic of skewed degree distribution. When the network size grows as large as in Facebook, Twitter, or a mobile phone network, the average clustering coefficient in a simple random graph approaches zero and the geodesic distance between any two vertices approaches infinity.

This is why we need to generate a random graph with clustering. Not only do we fix the number of vertices $|V|$ at the nodal level and the number of edges $|E|$ at the dyadic level, but we push the ordinary configuration model to go beyond the dyadic level by fixing the average clustering coefficient $C(G)$ and/or global clustering coefficient C_Δ at the triadic level. In this way we can reproduce the characteristics of non-zero clustering coefficient and limited geodesic distance as in the real-world networks. These two characteristics are also associated with other important network properties such as community structure and the existence and evolution of giant component. It will also enable us to study network robustness, percolation properties, cascading failure, the diffusion process, and the effect of network topology on the dynamical systems.

3 Four existing algorithms for generating clustered random graph

There are at least four existing algorithms to generate random graph with clustering advanced in recent years all of which should give credit to the pioneering works of Serrano and Boguñá (2005, 2006a, 2006b). Based on the working processes, these four algorithms can be summarized into two groups: adding triangles to given random networks by rewiring edges, and generating triangles based on given models.

[1] Triangle refers to a network structure of three vertices which connect with one another.

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

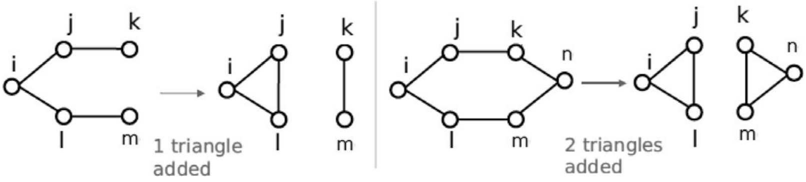


Fig. 1. Adding triangle(s) by rewiring edges (Source: Bansal et al. 2009).

3.1 Adding triangles to given random networks by rewiring edges

The first two algorithms start from given random networks. In the algorithm of Guo and Kraines (2009), it is a simple random graph G with a set of vertices V and a set of edges E following a given degree sequences as in the configuration model (see Molloy & Reed, 1995). In the algorithm of Bansal et al. (2009), it is a real network rewired to be completely random.

As shown in Figure 1, triangles are added to the random networks in two ways: i) a chain of five vertices k, j, i, l, and m is randomly selected and one triangle is added at a time by rewiring edges e_{jk} and e_{lm} to e_{jl} and e_{km}; and ii) a ring of six vertices i, j, k, n, m, and l is randomly selected and two triangles are added at a time by rewiring edges e_{jk} and e_{lm} to e_{jl} and e_{km}.

The rewiring process is repeated until we get the same global clustering coefficient C_{Δ} and/or average clustering coefficient $C(G)$ as in the real network, or it reaches a certain predefined number of trials (Guo & Kraines, 2009; Bansal et al., 2009).

3.2 Generating triangles based on given models

A model is given to generate a clustered random graph in the latter two algorithms. In Newman-Miller algorithm, it is a configuration model $G(V, S, T)$ which defines the number of vertices $|V|$, the number of single edges $|S|$, and the number of triangles $|T|$ (Newman, 2009; Miller, 2009). In Gleeson's algorithm (2009), it is a joint degree distribution $\gamma_{d_i,k}$model specifying the probability a vertices i has degree d_i and is part of a k-clique.

As shown in Figure 2 (left), in Newman-Miller algorithm, a triangle is added by joining three vertices at random and this process is repeated until all the vertices are parts of some unique triangles. A single edge is added by joining two vertices at random and this process is repeated until all the vertices are parts of some unique single edges (Newman, 2009). Gleeson (2009) generalized the Newman-Miller algorithm by using higher-order motif – a k-clique, which is a complete graph among k vertices each of which is connected to every other vertex in the graph, and the author used external link (which is similar to Newman's single edge and represents the edges not involved in any cliques) to join all the k-cliques together. For example, if the mean degree of a real network is between 3 and 4, a clustered random graph can be generated by joining some 3-cliques (triangles), some 4-cliques, and with the remainder as individuals (i.e., 1-cliques) as shown in Figure 2 (right)

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

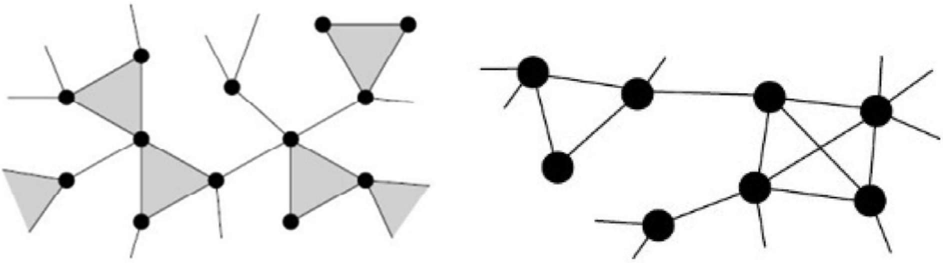


Fig. 2. Generating function. (Source: Newman 2009; Gleeson 2009)

4 A generalized version of Gleeson's algorithm

Guo and Kraines (2009) experimented their algorithm by generating a network with 1,000 vertices and 4,000 edges. Bansal et al. (2009) compared clustered random graphs with five real networks among which the maximum number of vertices was 4,713. Gleeson (2009) simulated networks of maximum size 10^5. In other words, none of those algorithms have been tested for large-scale networks such as Facebook, Twitter, or other large and complex communication networks data sets.

In this section we try to use those four algorithms to generate clustered random graphs for a large-scale mobile phone network of over 10 million subscribers of one unnamed mobile phone company.[2] The raw data provide details of time, origins, call types, destinations and durations. We focus on the voice-call communication behaviors during four weeks – from August 3, 2008 (Sunday) to August 30, 2008 (Saturday) and convert it to an undirected graph.

This piece of network data consist of 6,719,330 active vertices[3] and 15,913,611 edges[4]. And the average nodal degree is $\overline{d} = \frac{|E|}{|V|} = \frac{15,913,611}{6,719,330} = 2.37$ and the network density is $\Delta = \frac{2|E|}{|V|(|V|-1)} = \frac{2\times 15,913,611}{6,719,330\times(6,719,330-1)} = 7.05e - 7$.

At the triadic level, there are 126,175,382 2-paths among which 109,383,149 are structural holes[5] and 5,597,411 are triangles. The average clustering coefficient $C(G)$is 0.24, and the global/overall clustering coefficient C_Δ is 0.13.

Here we should notice that the global clustering coefficient C_Δ is more appropriate a target indicator of clustering for a large-scale network with millions of vertices and tens of millions of edges. The average clustering coefficient $C(G)$ works fine for a small network with hundreds or thousands of vertices (i.e., Guo & Kraines, 2009), but is not efficient for edge rewiring jobs as described in the first two algorithms

[2] The network data have been used in numerous publications (see Bagrow et al., 2011; Ercsey-Ravasz et al., 2011; Ghoshal & Barabási, 2011; Hidalgo & Rodriguez-Sickert, 2008; Lichtenwalter et al., 2010; Liu et al., 2011; Onnela et al., 2011; Raeder et al., 2011; Wang et al., 2011; Wang et al., 2013).

[3] The total number of customers is about 10 million and about 6.7 million of them were active (that is, having at least one communication behavior) during the four weeks.

[4] In the undirected graph the relationship between any two vertices i and j is symmetric $e_{ij} = = e_{ji}$, and as a result we can use either double counting – both e_{ij} and e_{ji} are included in the edge list – or single counting – only one of e_{ij} and e_{ji} is included in the edge list – and the number of edges $|E|$ in the former strategy is twice as that of the latter one. In this study we adopt single counting and all the calculations are adjusted for this situation.

[5] 2-path refers to a network structure that an ego has two alters. If these two alters are connected, it is a triangle; and if not, it is a structural hole. Structural hole is first advanced by Burt (1995) and refers to a structure that an ego has two alters who does not connect with each other.

since it will take an unacceptably long time to update the triangle list and 2-path list millions of times for a large-scale network.

By adopting the algorithm of Guo and Kraines and that of Bansal et al., two groups of clustered random graphs having the same number of vertices, edges, and global clustering coefficient as in the mobile phone network are successfully generated. For the first group, the average clustering coefficient of the clustered random graph is 0.22, which is a little bit smaller than that in the real network 0.24; and in the second group, the average clustering coefficient of the clustered random graph is 0.41, which is much larger than that in the real network. The rewiring processes through the algorithm of Guo and Kraines take about 490 hours, and those through the algorithm of Bansal et al. take about 3,150 hours[6].

Newman-Miller algorithm is performed in two steps: the first, randomly connecting three vertices to fit the expect number of triangles, and this step takes about 3.5 hours; and the second, generating single edges among the triangles, which turns out to be impossible. The problem lies in the fact that it over-uses the edges to produce the same number of triangles as in the real network – in the real network the 5,597,411 triangles only use up 8,474,226 edges (about 53.25% of all edges), while by adopting Newman-Miller algorithm the 5,597,411 triangles use 15,171,585 edges (about 95.34% of all edges) and there are only 742,026 edges left for single edges, which means there are not enough structural holes being generated. Thus the Newman-Miller algorithm fails to fit the global clustering coefficient C_Δ as in the real network.

Gleeson's algorithm seems to go to the flipped side. Instead of over-using edges to generate a certain amount of triangles as in Newman-Miller algorithm, Gleeson's algorithm over-produces triangles with certain number of edges through the combination of k-cliques.

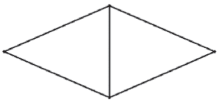
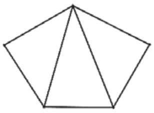
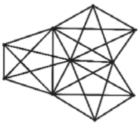
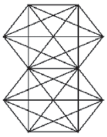
The only way out is that we should not constrain ourselves on k-cliques. We can turn to combination of other motifs which have the following structures: i) there are three or more vertices in the motif; ii) edges in the motif are not completely connected as in a k-clique; and iii) therefore there are both triangles and structural holes in the motif. The configuration model is extended as $G(V_S, V_T, E_S, E_T, S, T)$, where V_S and V_T represent the single-degree vertex set (i.e., isolate) and the multiple-degree vertex set, Es and E_T represent the external links between motifs to form structural holes and the edge set within motifs to generate triangles as well as structural holes, and S and T represent the structural hole vector and the triangle vector.

In this way we generalize Gleeson's algorithm which is executed in two steps. Step 1, the triangles are generated by V_T, E_T, and T as in the real network. For example, we can suppose the expected clustered random graph is composed of four motifs as shown in Table 1 – *a*) two triangles sharing a common edge, *b*) three triangles in a pentagon sharing a common vertex, *c*) three 5-cliques sharing a common vertex, and *d*) two 6-cliques sharing a common edge.

[6] We test these algorithms on a server with Linux 2.6.18–274.12.1.e15 operating system, two Intel Xeon X5450 3.00GHz 4-core CPUs, 64GB DDR2 667MHz PC2–5300 RAM, and twelve Western Digital WD1001FALS-0 hard drives (7200 RPM, 1TB, 32MB cache) in a RAID 60 array.

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

Table 1. *A clustered random graph formed by linking four motifs: a) two triangles sharing a common edge, b) three triangles in a pentagon sharing a common vertex, c) three 5-cliques sharing a common vertex, and d) two 6-cliques sharing a common edge.*

	Motif a	Motif b	Motif c	Motif d
				
# of vertices	4	5	10	10
# of edges	5	7	27	29
# of triangles	2	3	31	40
# of structural holes	2	5	42	32

And the expected clustered random graph should fit the following equations

$$\begin{cases} \text{nodes}: & 4x+5y+10z+10w = 5,358,175 \ (V_T) \\ \text{edges}: & 5x+7y+27z+29w = 8,474,226 \ (E_T) \\ \text{triangles}: & 2x+3y+31z+40w = 5,597,411 \ (T) \end{cases}$$

If we force that the number of motif c and that of motif d to be equal, we get

$$\begin{cases} x = 373,255 \\ \mathrm{y} = 601,383 \\ z = 42,912 \\ w = 42,912 \end{cases}$$

And step 2, external links are added to generate the left-over structural holes. There are already $2x+5y+42z+32w = 6,928,913$ structural holes within motifs, and we need 102,454,236 = 109,383,149 – 6,928,913 more structural holes by adding 7,439,385 external links between motifs, which mean on average each external link generate 13.77 = 102,454,236/7,439,385 structural holes. And since the greater-nodal-degree vertices are located in motif c and d, we assign half external links between motif c and d, and one quarter each between motif a and b and between motif b and c.

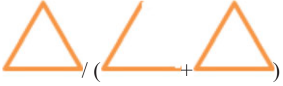
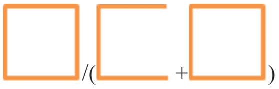
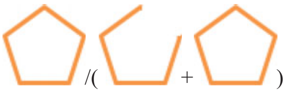
It takes about 9.1 hours to get one expected random graph with clustering. The global clustering coefficient C_Δ is 0.13, which is the same as in the mobile phone network. The average clustering coefficient of the clustered random graph is 0.35, which is greater than that in the real network 0.24.

5 Randomness evaluation of the algorithms for generating clustered random graphs

The generalized version of Gleeson's algorithm $G(V_S, V_T, E_S, E_T, S, T)$ fixes network properties at the nodal, dyadic, and triadic level, and thus we need go even higher levels (i.e., the tetradic and pentadic levels) to see how random the clustered random graph is. The network density Δ in is used as the randomness evaluation indicator.

As shown in Table 2, in the initiated random graph G generated for the edge rewiring processes of the algorithm of Guo and Kraines, at the triadic level the global clustering coefficient C_Δ is 7.50e-7 which is very close to the network density

Table 2. *Probability of edge closure at the tetradic and pentadic levels.*

	Initiated random graph G	Clustered random graph by algorithm of Guo and Kraines	Clustered random graph by algorithm of Bansal et al.	Clustered random graph by the generalized version of Gleeson's algorithm
Triadic level (C_{Δ}/ $T(G)$) 	7.50e-7	0.13	0.13	0.13
Tetradic level 	7.13e-7	5.15e-4	4.17e-4	2.66e-6
Pentadic level 	5.69e-7	3.87e-5	3.83e-5	6.44e-6

7.05*e*-7, and both the tetradic closure and pentadic closure ratios are at the *e*-7 level, which confirms that at higher-order network levels this graph is completely random.

By adopting the algorithm of Guo and Kraines, the algorithm of Bansal et al, and the generalized version of Gleeson's algorithm, three groups of clustered random graphs are generated. As shown in Table 2, the tetradic closure and pentadic closure ratios in the clustered random graph generated by the generalized version of Gleeson's algorithm are at the 10^{-6} level which much closer to the network density than those generated by the other two algorithms. Therefore the graph generated by the generalized version of Gleeson's algorithm is relatively more random than the other two.

6 Application to other large-scale networks

Next the generalized version of Gleeson's algorithm $G(V_S, V_T, E_S, E_T, S, T)$ is applied to generate clustered random graphs for other large-scale networks. There are eight large-scale network data sets listed in Table 3 all of which comes from the Stanford Large Network Dataset Collection at http://snap.stanford.edu/data/[7]. Those network data share some common characteristics with the large-scale mobile

[7] There are over seventy network data sets available from the webpage. Three types of network data sets are skipped, including those: i) network size are relatively small (i.e., the social circles from Facebook & Wikipedia who-votes-on-whom network), ii) numbers of 2-paths exceed 2.1 trillion, the maximum matrix length the server can handle (i.e., the LiveJournal online social network & the YouTube online social network), and iii) over the server's memory (i.e., the Orkut online social network & the 476 million tweets data set). And finally eight network data sets are selected. All the networks are converted to undirected graph before applying the algorithm.

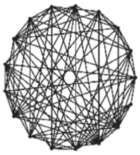
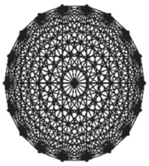
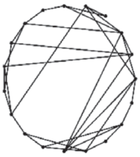
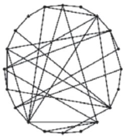
Table 3. *Large-scale network data sets from the Stanford Large Network Dataset Collection.*

	Patent citation network	Amazon product co-purchasing network	DBLP collaboration network	Epinions social network
V_S: single-degree vertices	667,336	25,709	43,181	67,390
V_T: multiple-degree vertices	3,107,432	309,154	273,899	64,190
E_S :external links	8,725,041	211,212	73,142	158,430
E_T: edges in triangles	7,793,906	714,660	976,724	552,780
S: number of structural holes	313,236,204	7,750,799	15,107,734	167,463,239
T: number of triangles	7,515,023	667,129	2,224,385	4,910,076
Δ: density	2.32e-6	1.65e-5	2.09e-5	8.22e-5
$C(G)$: average clustering coefficient	0.09	0.43	0.73	0.26
C_Δ: global clustering coefficient	0.07	0.21	0.31	0.08
Sources	Leskovec et al. (2005)	Yang & Leskovec (2012)	Yang & Leskovec (2012)	Leskovec et al. (2010)

	Flickr image relationships	Gowalla	Google web graph	Notre Dame web graph
V_S: single-degree vertices	313	49,452	153,407	161,832
V_T: multiple-degree vertices	105,625	147,139	722,306	163,897
E_S :external links	364,257	207,631	478,521	294,706
E_T: edges in triangles	1,952,691	742,696	3,843,530	795,402
S: number of structural holes	482,716,716	283,580,626	687,241,515	278,151,159
T: number of triangles	107,987,357	2,273,138	13,391,903	8,910,005
Δ: density	4.13e-4	4.92e-5	1.13e-5	2.05e-5
$C(G)$: average clustering coefficient	0.09	0.32	0.62	0.47
C_Δ: global clustering coefficient	0.40	0.02	0.06	0.09
Sources	McAuley & Leskovec (2012)	Cho et al. (2011)	Leskovec et al. (2009)	Albert et al. (2009)

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

Table 4. *One possible motif solution for the patent citation network.*

	Motif a	Motif b	Motif c	Motif d
				
# of vertices	17	18	19	20
# of edges	86	153	33	38
# of triangles	166	816	5	7
# of structural holes	314	0	33	107
# of motifs	37,299	694	63,099	63,099

network data: i) the network density is relatively low when compared with small-size networks; ii) on average each edge in E_T helps generate more than one triangle and in extreme cases (i.e., Flickr image relationships) each edge is located in more than 50 triangles; and iii) on average each external link in E_S helps generate at least 10 structural holes and in extreme cases (i.e., the latter five networks in Table 3) each external link is required to generate more than 900 structural holes.

The generalized version of Gleeson's algorithm successfully generates clustered random graph for those eight large-scale networks. For example, one possible motif solution for the patent citation network is given in Table 4.

Turning to randomness evaluation, since it takes weeks and months to generate clustered random graphs using the algorithm of Guo and Kraines and that of Bansal et al., we select three out of eight networks which have relatively fewer triangles and thus need fewer edge rewiring steps. As shown in Table 5, the generalization version of the Gleeson's algorithm still performs better in randomness evaluation and computing time than the other two algorithms.

7 Conclusions

Random graphs are commonly used to compare with real networks. However, a simple random graph in large-scale often lacks of local structure beyond the dyadic level and as a result we need to generate the clustered random graph to compare the local structure at higher-order network levels.

As shown in Table 6, we successfully generate three groups of clustering random graphs in which the global clustering coefficient C_Δ as well as the number of vertices $|V|$ and the number of edges $|E|$ are the same as in the real networks based on the algorithm of Guo and Kraines, the algorithm of Bansal et al., and the generalized version of Gleeson's algorithm. The Newman-Miller algorithm doesn't work because it over-uses edges to generate the same number of triangles as in the real network and thus both the number of structural holes and the global clustering coefficient are not kept.

And by comparing the tetradic closure and pentadic closure ratios, the clustered random graph generated by our generalized version of Gleeson's algorithm seems

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

Table 5. *Tetradic closure and pentadic closure ratios for three large-scale networks.*

	Initiated random graph *G*	Clustered random graph by algorithm of Guo and Kraines	Clustered random graph by algorithm of Bansal et al.	Clustered random graph by the generalized version of Gleeson's algorithm
Amazon product co-purchasing network				
Computing time (hours)		65	393	4.5
Tetradic closure	1.42e-5	7.07e-3	5.25e-3	5.29e-4
Pentadic closure	1.27e-5	8.16e-3	7.33e-3	5.87e-4
DBLP collaboration network				
Computing time (hours)		257	1,282	5.7
Tetradic closure	1.96e-5	9.25e-3	8.89e-3	4.07e-4
Pentadic closure	1.94e-5	3.07e-3	3.56e-3	7.77e-4
Gowalla				
Computing time (hours)		266	1,311	6.2
Tetradic closure	4.87e-5	6.09e-2	5.44e-2	7.25e-4
Pentadic closure	2.17e-5	2.67e-2	2.83e-2	8.06e-4

Table 6. *Algorithm summary for generating large-scale clustered random graphs.*

		Algorithm of Guo and Kraines	Algorithm of Bansal et al.	Newman-Miller algorithm	The generalized version of Gleeson's algorithm
Nodal level	# of vertices	√	√	√	√
	nodal degree for each vertex[8]	×	√	×	×
Dyadic level	# of edges	√	√	√	√
	Average nodal degree	√	√	√	√
	Network density	√	√	√	√
Triadic level	# of 2-paths	×	√	×	√
	# of structural holes	×	√	×	√
	# of triangles	×	√	√	√
	Global clustering coefficient	√	√	×	√
	Average clustering coefficient	×	×	×	×

to be more random than those generated by the algorithm of Guo and Kraines and the algorithm of Bansal et al.

Another advantage of our generalized version of Gleeson's algorithm is its computation efficiency. While it takes weeks and months to get a clustered random

[8] To preserve the nodal degree of each vertex as in the real network is necessary for the algorithm of Bansal et al. In this way the number of 2-paths is fixed and we can just keep rewiring until we get the expected number of closed 2-paths – triangles. But it is not necessary for the Newman's algorithm and the generalized version of Gleeson's algorithm which give models to reproduce the expected numbers of triangles and structural holes.

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

graph with the first two algorithms, we can generate a clustered random graph based on our generalized version of Gleeson's algorithm usually in several hours.[9]

One criticism to Gleeson's algorithm is that it might not be a random process that vertices are set to be clustered in k-cliques to generate triangles. This critique could also be applied to the generalized version of Gleeson's algorithm – it might not be a random process that vertices are set to be clustered in motifs to generated triangles, and of course also applied to its specific version – Newman-Miller algorithm. However, from this perspective the first two algorithms do not have any advantage since the edge rewiring processes might not be completely random either.

From our point of view, as long as we fix the global clustering coefficient as in real networks at the triadic level, the generation processes are no longer as random as supposed in the critique. What we can assure is that each vertex has the same opportunity to be assigned to a configuration – a triangle as in Newman-Miller algorithm, a k-clique as in Gleeson's algorithm, and a motif as in our generalized version of Gleeson's algorithm – or to a rewiring process as in the algorithm of Guo and Kraines and that of Bansal et al. At the even higher levels (i.e., tetradic and pentadic levels), the tie formation process is inclined to be random.

References

Albert, R., Jeong, H., & Barabási, A.-L. (1999). Diameter of the World-Wide Web. *Nature*, **401**, 130–131.

Bagrow, J. P., Wang, D., & Barabási, A.-L. (2011). Collective response on human populations to large-scale emergencies. *PLoS One*, **6**, 1–8.

Bansal, S., Khandelwal, S., & Meyers, L. A. (2009). Exploring biological network structure with clustered random networks. *BMC Bioinformatics*, **10**, 405–419.

Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, **286**, 509–512.

Burt, R. S. (1995). *Structural Holes: The Social Structure of Competition*. Cambridge, MA: Harvard University Press.

Cartwright, D. & Harary, F. (1956). Structural balance: A generalization of Heider's theory. *Psychological Review*, **63**, 277–293.

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, pp. 1082–1090.

Davis, J. A. (1967). Clustering and structural balance in graphs. *Human Relations*, **30**, 181–187.

Davis, J. A. (1979). The Davis /Holland /Leinhardt studies: An overview. In P. W. Holland, & S. Leinhardt (Eds.), *Perspectives on Social Network Research* (pp. 51–62). New York: Academic Press.

de Sola Pool, I., & Kochen, M. (1978/1979). Contacts and influence. *Social Networks*, **1**, 5–51.

Ercsey-Ravasz, M., Lichtenwalter, R. N., Chawla, N. V., & Toroczkai, Z. (2011). Range-limited Centrality Measures in Non-weighted and Weighted Complex Networks, *arXiv* e-print, **1111.5382**.

Erdős, P., & Rényi, A. (1959). On random graphs I. *Publicationes Mathematicae*, **6**, 290–297.

Ghoshal, G., & Barabási, A.-L. (2011). Ranking stability and super-stable nodes in complex networks. *Nature Communications*, **2**, 1–7.

[9] In extreme case such as the Flickr image relationships its clustered random graph was generated in 48 hours.

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

Gleeson, J. P. (2009). Bond percolation on a class of clustered random networks. *Physical Review E*, **80**, 036107.

Granovetter, M. (1973). The strength of weak ties. *American Journal of Sociology*, **78**, 1360–1380.

Guo, W., & Kraines, S. B. (2009). A random network generator with finely tunable clustering coefficient for small-world social networks. *Proceedings of the 2009 International Conference on Computational Aspects of Social Networks*. Washington, DC: IEEE Computer Society, pp. 10–17.

Heider, F. (1946). Attitudes and cognitive organization. *Journal of Psychology*, **21**, 107–112.

Hidalgo, C. A., & Rodriguez-Sickert, C. (2008). The Dynamics of a mobile phone network. *Physica A*, **387**, 3017–3024.

Krackhardt, D. (1998). Simmelian ties: Super strong and sticky. In R. M. Kramer & M. A. Neale (Eds.), *Power and Influence in Organizations* (pp. 21–38). Thousand Oaks, CA: Sage.

Krackhardt, D., & Handcock, M. S. (2006). Heider vs. Simmel: Emergent features in dynamic structures. In E. M. Airoldi, & D. M. Blei (Eds.), *Statistical Network Analysis: Models, Issues and New Directions (ICML 2006)* (pp. 14–27). Berlin: Springer.

Leskovec, J., Huttenlocher, D., & Kleinberg, J. (2010). Signed networks in social media. *CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 1361–1370.

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2005). Graphs over time: Densification laws, shrinking diameters and possible explanations. In *KDD '05 Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. New York. ACM, pp. 177–187.

Leskovec, J., Lang, K., Dasgupta, A., & Mahoney, M. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, **6**, 29–123.

Lichtenwalter, R. N., Lussier, J. T., & Chawla, N. V. (2010). New perspectives and methods in link prediction. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. New York: ACM, pp. 243–252.

Liu, Y.-Y., Slotine, J.-J., & Barabási, A.-L. (2011). Controllability of complex networks. *Nature*, **473**, 167–173.

McAuley, J., & Leskovec, J. (2012). Image labeling on a network: Using social-network metadata for image classification. *ECCV'12 Proceedings of the 12th European conference on Computer Vision - Volume Part IV*, Berlin, Heidelberg: Springer-Verlag, pp. 828–841.

Miller, J. C. (2009). Percolation and epidemics in random clustered networks. *Physical Review E*, **80**, 020901(R).

Molloy, M., & Reed, B. (1995). A critical point for random graphs with a given degree sequence. *Random Structures & Algorithm*, **6**, 161–179.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, **45**, 167–256.

Newman, M. E. J. (2009). Random graphs with clustering. *Physical Review Letters*, **103**, 05870.

Onnela, J. P., Arbesman, S., Gonzalez, M. C., Barabási, A.-L., & Christakis, N. A. (2011). Geographic constraints on social network groups. *PLoS One*, **6**, 1–7.

Opsahl, T., & Panzarasa, P. (2009). Clustering in weighted networks. *Social Networks*, **31**, 155–163.

Raeder, T., Lizardo, O., Hachen, D., & Chawla, N. V. (2011). Predictors of short-term decay of cell phone contacts in a large-scale communication network. *Social Networks*, **33**, 245–257.

Serrano, M. Á., & Boguñá, M. (2005). Tuning clustering in random networks with arbitrary degree distributions. *Physical Review E*, **72**, 036133.

Serrano, M. Á., & Boguñá, M. (2006a). Clustering in complex networks. I. General formalism. *Physical Review E*, **74**, 056114.

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press

Serrano, M. Á., & Boguñá, M. (2006b). Clustering in complex networks. II. Percolation properties. *Physical Review E*, **74**, 056115.

Simmel, G. (1908/1950). *The Sociology of Georg Simmel*. New York: Free Press.

Travers, J., & Milgram, S. (1969). An experimental study of the small world problem. *Sociometry*, **32**, 425–443.

Wang, C., Lizardo, O, Hachen, D., Strathman, A., Toroczkai, Z., & Chawla, N. V. (2013). A dyadic reciprocity index for repeated interaction networks. *Network Science*, **1**, 31–48.

Wang, D., Pedreschi, D., Song, C, Giannotti, F., & Barabási, A.-L. (2011). Human mobility, social ties, and link prediction. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, New York: ACM, pp. 1100–1108.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of "Small-world" networks. *Nature*, **393**, 440–442.

Yang, J., & Leskovec, J. (2012). Defining and evaluating network communities based on ground-truth. *MDS '12 Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, Article No. 3. New York: ACM.

https://doi.org/10.1017/nws.2014.7 Published online by Cambridge University Press