Constrained graph generation: Preserving diameter and clustering coefficient simultaneously

Dávid Ferenczi[1 [0009−0000−9507−5774]],
Alexander Grigoriev[1 [0000−0002−8391−235X]]

Maastricht University, Maastricht, 6211LM, Netherlands
{d.ferenczi,a.grigoriev}@maastrichtuniversity.nl

Abstract. Generating graphs subject to strict structural constraints is a fundamental computational challenge in network science. Simultaneously preserving interacting properties, e.g., the diameter and the clustering coefficient, is particularly demanding. Simple constructive algorithms often fail to locate vanishingly small sets of feasible graphs, while traditional Markov-Chain Monte Carlo (MCMC) samplers suffer from severe ergodicity breaking. In this paper, we propose a two-step hybrid framework combining Ant Colony Optimization (ACO) and MCMC sampling to overcome the aforementioned limitations.

Keywords: Graph generation · ACO · MCMC

1 Introduction

Generating graphs with prescribed properties is a well-known problem at the intersection of random graph theory [4,12] and network science [14,11]. From a theoretical perspective, inquiry focuses on constructive existence: Can a member of a graph family $\mathcal{G}$ be generated in polynomial time?

For properties that can be checked in polynomial time, Markov-Chain Monte Carlo (MCMC) methods can be used [15], but these methods often fail due to the lack of ergodicity [5]. To overcome this limitation, we turn to using Ant Colony Optimization (ACO) algorithms [3,6] for their ability to explore $\mathcal{G}$ efficiently by constructing graphs one edge at a time. We utilize ACO-generated graphs as diverse seeds for MCMC. In the nutshell, our two step framework combines ACO to explore $\mathcal{G}$ by finding various seed graphs, and then using MCMC for sampling, yet preserving the desired properties of generated graphs.

In this paper we want to sample graphs with a given number of nodes and edges maintaining the diameter and clustering coefficient within parametrized bounds. Notice, these parameters strongly interact with each other: increasing clustering coefficient creates structural redundancy decreasing the diameter of the graph [17,13]. In applied network analysis, generating ensembles of such strictly constrained graphs is essential for creating rigorous null models allowing researchers to test the statistical significance of complex phenomena observed

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

in real-world systems against baselines that preserve both local transitivity and global path lengths.

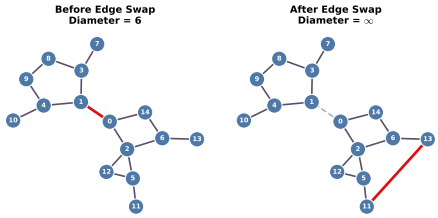

Fig. 1. Impact of an edge swap on diameter. Moving a single red edge from a central position (Left) to a boundary position (Right) fundamentally alters the diameter.

2 Problem statement and preliminaries

Let $G = (V, E)$ be a simple graph on n nodes and m edges. A *triplet* in a graph is 3 nodes connected by two edges, and a *closed triplet* is a graph on 3 nodes, that forms a triangle. If there is at least one triplet in the graph, we refer to $\mathrm{gcc}(G) = \frac{\text{number of closed triplets}}{\text{number of all triplets}}$ as the *global clustering coefficient* and we let $\mathrm{gcc}(G) = 0$ if there are no triplets in the graph. In the literature, this quantity is also referred as *transitivity* or *global clustering coefficient*. The number of closed triplets is 3 times the number of triangles [9]. We will use the notation Δ_G and Λ_G for the number of triangles and triplets, respectively.

The *diameter* of G is defined as $\mathrm{diam}(G) = \max_{u,v \in V} d(u, v)$ and the *eccentricity* of a node $u \in V$ is $\mathrm{ecc}(u) = \max_{v \in V \setminus \{u\}} d(u, v)$, where $d(u, v)$ is the length of the shortest path connecting u and v. Using the eccentricity we can state the following well-known bound for the diameter [9]: $\mathrm{ecc}(v) \leq \mathrm{diam}(G) \leq 2 \cdot \mathrm{ecc}(v)$ for any node $v \in V$.

Now we formally state the problem. Given the number of nodes n, the number of edges m, and intervals $[\mathrm{gcc}_{\min}, \mathrm{gcc}_{\max}]$, $[\mathrm{diam}_{\min}, \mathrm{diam}_{\max}]$, we want to generate a set of graphs $(G_i)_{i=1}^{\ell}$, such that $\mathrm{gcc}(G_i) \in [\mathrm{gcc}_{\min}, \mathrm{gcc}_{\max}]$ and $\mathrm{diam}(G_i) \in [\mathrm{diam}_{\min}, \mathrm{diam}_{\max}]$.

2.1 Simple edge swaps

An *edge swap* is simultaneous deletion of an edge between two connected nodes u and v, and insertion of an edge between two disconnected nodes x and y. Such an operation is often used in graph rewiring algorithms [2]. We can see in Figure 1, that the diameter can behave erratically under simple edge swaps.

For an arbitrary node $x \in V$, we call the set of nodes $N_x = \{u | (u, x) \in E\}$ the neighborhood of x. For the degree of node x, we use the notation δ_x. Note that a naive calculation of $\mathrm{gcc}(G)$ would take $\mathcal{O}(n\langle\delta\rangle^2)$ time, where $\langle\delta\rangle$ is the expected degree in the graph. However, for every swap operation in a graph,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

we can update the value $\mathrm{gcc}(G)$ in $\mathcal{O}(\delta_{max})$ time as the changes to the triangle count are strictly local.

Incremental update rule for gcc. An edge swap between u, v and x, y alters the triangle count Δ_G and triplet count Λ_G locally. Specifically, $\Delta_{\mathrm{new}}(G) = \Delta(G) - |N_u \cap N_v| + |N_x \cap N_y|$, and $\Lambda_{\mathrm{new}} = \Lambda(G) + \delta_x + \delta_y - (\delta_u + \delta_v - 2)$. The updated clustering coefficient is then: $\mathrm{gcc}(G') = 3\Delta_{new}/\Lambda_{new}$.

Effect on the diameter. Due to the global nature of the diameter, there are no guaranteed bounds for the change of the diameter under edge swaps. Recalculation of the diameter after each attempted edge swap requires an All-Pair-Shortest-Path algorithm, with runtime $\mathcal{O}(n(n+m))$. We can approximate the diameter using the Double-Sweep BFS heuristic [1], that runs in linear time, and outputs an estimator $\hat{D}$, which often coincides with the real diameter [10]. Combining $\hat{D}$ with the eccentricity inequality yields a strict range for the true diameter.

3 Algorithm overview

We continue by providing an overview of our algorithm.

First, we describe a Metropolis-Hastings algorithm in Subsection 3.1, that takes a graph as an input and rewires its edges, keeping n and m fixed, while diam and gcc within bounds. This is not a proper graph generation method, as it does not generate graphs from scratch, but it can rewire a given seed graph, and therefore it can serve for sampling.

To generate seeds, we introduce an ACO algorithm in subsection 3.2, which can construct graphs given a list of desired parameters. We combine the algorithms by running ACO first to get a set of graphs that can be used as an input for the MCMC to reshuffle, and therefore get uniformly distributed samples.

3.1 Metropolis-Hastings algorithm

By proposing edge swaps and accepting the new graphs if the new clustering coefficient and diameter are within an acceptable range, we define a rewiring algorithm that keeps both properties within required bounds.

This simple mechanism is in fact a Metropolis-Hastings algorithm, which can be used to sample graphs uniformly at random [5,7,16]. The steps of the procedure are detailed in Algorithm 1.

It can be shown that not every graph can be constructed by MCMC from a given seed graph by rewiring when using only simple edge swaps. This means that the Markov-chain is not ergodic, which is essential for our sample to be truly uniformly distributed. We overcome this limitation by using an ACO algorithm, that generates a set of different starting graphs, that will serve as different seeds for the MCMC.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

Algorithm 1 Metropolis-Hastings Step

Require: $(G_t, \Delta_t, \Lambda_t, C_{bounds}, D_{bounds})$
1: $(u, v) \leftarrow \text{rand}(E(G_t))$; $(x, y) \leftarrow \text{rand}(V^2 \setminus E(G_t))$
2: $G' \leftarrow (G_t \setminus \{(u, v)\}) \cup \{(x, y)\}$
3: Update Δ', Λ', C' via Eq. 1 ▷ Local check
4: **if** $C' \in [C_{\min}, C_{\max}]$ **and** DoubleSweepBFS$(G') \in [D_{\min}, D_{\max}]$ **then**
5: **return** (G', Δ', Λ') ▷ Accept
6: **else**
7: **return** $(G_t, \Delta_t, \Lambda_t)$ ▷ Reject
8: **end if**

3.2 ACO for Graph Construction

To resolve the ergodicity issue practically, we propose a hybrid framework, where ACO serves as a *constructive* global searcher [3,6]. Using this approcach, we can generate a broader variety of structurally different graphs, while keeping our constrained parameters in control.

ACO starts as a random search and governs itself in the direction of feasible solutions by learning from the mistakes and good choices it made while searching. In order to guarantee that the diameter remains within bounds, and exploit the constructive nature of ACO, we are using a layered constructing heuristic.

Layered construction. In order to guarantee a lower bound $\text{diam}_{\min}$ for the diameter, we split the set of nodes n into $\text{diam}_{min}+1$ groups $L_1, L_2, \ldots L_{\text{diam}_{\min+1}}$. A layer of nodes is a subset of vertices that constraints the edge formation in order to preserve the diameter. Edges are only allowed to be formed within layers and between neighboring layers. Formally, we create a set $\mathcal{E}_{\text{available}}$, consisting of allowed edges in the graph: $(u, v) \in \mathcal{E}_{\text{available}} \iff |\text{layer}(u) - \text{layer}(v)| \leq 1$.

Pheromones and edge probabilities. To continue building our ACO algorithm we need to specify how the pheromone scores affect ants preferences towards certain edges. Let τ_e be the pheromones corresponding to the edge $e = (i, j)$. The probability for an edge being chosen by an ant is given by

$$p_e = \frac{\tau_e}{\sum_{o \in \mathcal{E}_{available}} \tau_o}. \tag{1}$$

Notice, ACO is a repetitive process. In the beginning, the pheromones are assumed to be uniformly distributed, i.e., $\tau_e = 1$ if $e \in \mathcal{E}_{\text{available}}$, and 0 otherwise. At each step of the ACO algorithm, k ants try to construct a valid graph by picking M distinct edges from $\mathcal{E}_{\text{available}}$, where every edge is chosen according to the distribution described by Equation (1).

Once the k ants have constructed the graphs, we update the pheromone scores on the edges after evaluating the ants performance by using a reward function. We do so by calculating the realized diameter and clustering coefficient, and defining the reward function corresponding to the ℓ-th ant's graph as follows:

$$R_k = \begin{cases} \frac{\text{diam}_{\text{valid}}}{\varepsilon + |C^\star - \text{gcc}(G_k)|} & \text{if } \hat{D} \leq D_{max}, \\ \frac{\text{diam}_{\text{invalid}}}{\varepsilon + |C^\star - \text{gcc}(G_k)|} & \text{otherwise,} \end{cases}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

Algorithm 2 Layered ACO Graph Construction

Require: $n, m, C_{bounds}, D_{bounds}, T$ iterations, K ants
Initialize $\tau_{ij} \leftarrow 1, \forall (i,j) \in \mathcal{E}_{avail}$
for $t = 1 \dots T$ **do**
  for $k = 1 \dots K$ **do**
    Construct G_k by sampling m edges $e \in \mathcal{E}_{avail}$ with $p_e \propto \tau_e$
    Evaluate R_k using $\text{gcc}(G_k)$ and $\hat{D}(G_k)$
  end for
  $\tau_e \leftarrow (1-\rho)\tau_e + \sum_{a \in S_{elite}} R_a W_e^a$ ▷ Update via Eq. 1
end for
return $\text{argmax}_{G_k}(R_k)$

where $C^\star$ is $\frac{\text{gcc}_{\min} + \text{gcc}_{\max}}{2}$, $\hat{D}$ is the estimation of the diameter acquired by doing a double sweep BFS, and $\text{diam}_{\text{valid}}$, and $\text{diam}_{\text{invalid}}$ are constants used to reward ants that produce graphs with valid diameters. In our implementation, we use 1 and 0.1 for those constants. This reward function is used to update the pheromones on the edges to boost the formation and dissolution of triangles as follows.

Pheromone score update. We employ an elitist pheromone update strategy to make sure that the ants who get closer to valid graphs can steer the others to look for solutions. This means that once all the k graphs have been constructed we rank them based on the graphs reward score, and new pheromones are updated based on the elite ants, or, in other words, the graphs associated with the highest fitness values.

To guide the search, we use the fact that the formation of intra-layer edges boosts triangle generation, whereas formation of inter-layer edges shrinks the number of triangles. We use the parameter ρ for pheromone evaporation rate. We implemented $\rho = 0.1$. The pheromone update is given by $\tau_e \leftarrow (1-\rho)\tau_e + \sum_{a \in S_{\text{elite}}} R_a \cdot W_e^a$ where W_e^a modulates triangle formation based on the target C^*:

$$W_e^a = \begin{cases} B & \text{if } \text{gcc}(G_a) < C^* \text{ and } e \text{ is intra-layer,} \\ & \text{or } \text{gcc}(G_a) > C^* \text{ and } e \text{ is inter-layer,} \\ H & \text{otherwise,} \end{cases} \tag{2}$$

where parameters B and H correspond to boosting and hindering edge formation within or between layers. In our implementation, $B = 2$ and $H = 0.5$. The formal description of the procedure is presented in Algorithm 2.

3.3 Hybrid Framework: From ACO to MCMC

We propose a hybrid strategy that leverages the complementary strengths of ACO and MCMC approaches. We first utilize the ACO algorithm to generate a set of distinct, feasible graph instances satisfying the target requirements n, m, $\text{gcc}_{\min}$, $\text{gcc}_{\max}$, $\text{diam}_{\min}$ and $\text{diam}_{\max}$. Unlike random graph generation with prescribed number of edges, which may fail to locate a feasible region in highly

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

constrained spaces, ACO acts as a guided global search to provide valid starting points. By seeding the sampler with topologically diverse solutions from ACO, we ensure the MCMC explores multiple distinct regions in the solution space, thereby generating more representative graph ensemble.

4 Experimental results

In this section we will report some results on our ACO-MCMC hybrid approach. All code and data for reproducing results are available online[1].

4.1 Success ratio and structural diversity

First we demonstrate the effectiveness of our ACO-MCMC framework for generating structurally diverse graphs. We evaluate the *success ratio* and *structural diversity* for given pairs of diam and gcc for a graph on 40 nodes with various density. Figure 2 shows the success ratio and structural diversity for a fixed edge denstiy. For each set of parameters, 100 instances of ACO were run with the

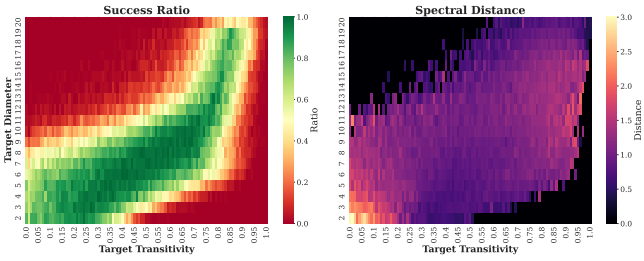

Fig. 2. Success ratio (left) and spectral distance (right) for graphs with edge density 0.2.

input parameters of the number of nodes, edges, gcc and diam. Each instance had 40 ants looking for solutions. *Success ratio* was calculated as the fraction of trials successfully finding graphs satisfying the given constraints.

To quantify the *structural diversity* of the generated graph ensembles, we use the *spectral distance* of the normalized Laplacian [19,8], normalized by the number of nodes. Let G be a graph obtained and let $\boldsymbol{\lambda}(G) = (\lambda_1, \lambda_2, \ldots, \lambda_N)$ denote the sequence of its normalized Laplacian eigenvalues sorted in non-decreasing order. The spectral distance of G and G' is $d_{\text{spectral}}(G, G') = \sqrt{\frac{\sum_{i=1}^{N}(\lambda_i - \lambda'_i)^2}{N}}$.

For a given set of constraints, we generate an ensemble of L valid graphs, denoted as $\mathcal{G} = \{G_1, G_2, \ldots, G_L\}$. To quantify the overall functional diversity of this ensemble, we calculate the mean pairwise spectral distance: $\mathcal{D}_{\text{ensemble}}(\mathcal{G}) = \frac{\sum_{1 \le i < j \le L} d_{\text{spectral}}(G_i, G_j)}{\binom{L}{2}}$. This metric represents the average structural variation

[1] `https://github.com/ferenczid/constrained_graph_generation`

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

within the solution space located by the algorithm. A lower value indicates graphs that are structurally similar, where high values show there are structurally different graphs satisfying the constraints [18].

4.2 Difference between MCMC and Hybrid MCMC

Next, we demonstrate the practical potential of our proposed hybrid framework by testing both, pure MCMC and hybrid MCMC, and measuring the structural difference of the seed and the output graphs of the MCMC and the hybrid MCMC. As shown in Figure 3 (**Left**), when the feasible space is heavily con-

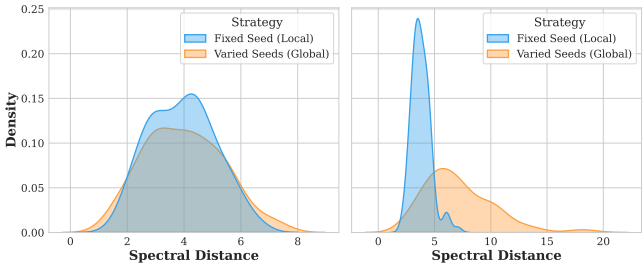

Fig. 3. Estimated density functions of the spectral distance (drift from seed). **Left:** ($n = 40$, $m = 78$, diam $= 12$, gcc $= 0.35$) Both methods find structurally similar solutions. **Right:** ($n = 40$, $m = 195$, diam $= 4$, gcc $= 0.4$) Varied seeds allow for more versatile sample.

strained, both methods converge to similar structures and their distributions overlap, indicating that standard MCMC is sufficient when the target topology is unique. However, in the case, where the constraints allow for a rich variety of valid graphs (**Right**)a clear divergence appears. While both methods find valid solutions, the standard MCMC remains trapped in the local neighborhood of its initialization (narrow peak). In contrast, the hybrid approach successfully samples from the broader solution space, capturing a much richer set of graphs.

5 Conclusion and future work

In this work, we developed and tested an efficient graph generation framework capable to keep in control clustering coefficient and diameter. While this framework already serves as a practical tool for generating valid graphs under given constraints, there are open questions that can be further investigated:

1. Is it possible to theoretically characterize graphs that can be reached from a seed doing edge swaps, while maintaining the constraints for clustering and the diameter?
2. How do competing constraints on local transitivity (clustering) and global path lengths (diameter) interact to shape the overall topology of the graph?
3. Alternatively, could soft-constraint MCMC methods serve as a viable, albeit computationally less efficient, alternative to our guided ACO initialization?

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51

References

1. Aingworth, D., Chekuri, C., Indyk, P., Motwani, R.: Fast estimation of diameter and shortest paths (without matrix multiplication). SIAM Journal on Computing (1999). https://doi.org/10.1137/S0097539796303421
2. Alstott, J., Klymko, C., Pyzza, P.B., Radcliffe, M.: Local rewiring algorithms to increase clustering and grow a small world. Journal of Complex Networks (2018). https://doi.org/10.1093/comnet/cny032
3. Blum, C.: Ant colony optimization: Introduction and recent trends. Physics of Life Reviews (2005). https://doi.org/https://doi.org/10.1016/j.plrev.2005.10.001
4. Bollobás, B.: Random Graphs. Cambridge Studies in Advanced Mathematics, Cambridge University Press (2001)
5. Brooks, S., Gelman, A., Jones, G., Meng, X.L.: Handbook of Markov chain Monte Carlo. CRC press (2011). https://doi.org/10.1201/b10905
6. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Computational Intelligence Magazine (2006). https://doi.org/10.1109/MCI.2006.329691
7. Earl, D.J., Deem, M.W.: Parallel tempering: Theory, applications, and new perspectives. Phys. Chem. Chem. Phys. (2005). https://doi.org/10.1039/B509983H
8. Gu, J., Hua, B., Liu, S.: Spectral distances on graphs. Discrete Applied Mathematics (2015). https://doi.org/https://doi.org/10.1016/j.dam.2015.04.011
9. Hofstad, R.v.d.: Random Graphs and Complex Networks. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press (2016)
10. Magnien, C., Latapy, M., Habib, M.: Fast computation of empirically tight bounds for the diameter of massive graphs. Journal of Experimental Algorithmics (04 2009). https://doi.org/10.1145/1412228.1455266
11. Maslov, S., Sneppen, K.: Specificity and stability in topology of protein networks. Science (2002). https://doi.org/10.1126/science.1065103
12. Molloy, M., Reed, B.: A critical point for random graphs with a given degree sequence. Random Structures & Algorithms (1995). https://doi.org/10.1002/rsa.3240060204, https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.3240060204
13. Newman, M.E.J.: The structure and function of complex networks. SIAM Review (2003). https://doi.org/10.1137/S003614450342480
14. Newman, M.: Networks (07 2018). https://doi.org/10.1093/oso/9780198805090.001.0001
15. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer Texts in Statistics, Springer, New York, NY, 2 edn. (2004). https://doi.org/10.1007/978-1-4757-4145-2
16. Sambridge, M.: A parallel tempering algorithm for probabilistic sampling and multimodal optimization. Geophysical Journal International (2014)
17. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature (1998). https://doi.org/10.1038/30918
18. Wills, P., Meyer, F.G.: Metrics for graph comparison: A practitioner's guide. PLOS ONE (2020). https://doi.org/10.1371/journal.pone.0228728
19. Wilson, R.C., Zhu, P.: A study of graph spectra for comparing graphs and trees. Pattern Recognition (2008). https://doi.org/https://doi.org/10.1016/j.patcog.2008.03.011

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_51