Random Graphs with Prescribed Degrees

Bogdan Grechuk · 8 Nov 2026

https://functor.network/user/3333/entry/2056

In September 2026, OpenAI released a preprint (OpenAI 2026) announcing a proof of the Kannan–Tetali–Vempala conjecture for simple undirected graphs. The conjecture concerns a natural problem at the intersection of graph theory, probability, and computer science: *how can we efficiently generate a uniformly random graph with prescribed vertex degrees?*

Let us begin with the definitions.

A *graph* consists of a set of vertices, some pairs of which are joined by edges. A graph is called *simple* if it has no loops or multiple edges.

The *degree* of a vertex is the number of its neighbours. If a graph has vertices labelled $1, 2, \dots, n$, its *degree vector* is

$$\mathbf{d} = (d_1, d_2, \dots, d_n),$$

where d_i is the degree of vertex i.

A degree vector is called *graphical* if there exists at least one simple graph having exactly these degrees.

For example, the vector

$$(1, 1, 1, 1)$$

is graphical. There are exactly three simple graphs on the labelled vertices $1, 2, 3, 4$ with these degrees: their edge sets are

$$\{\{1,2\},\{3,4\}\},$$
$$\{\{1,3\},\{2,4\}\},$$
$$\{\{1,4\},\{2,3\}\}.$$

Suppose we want to select one of these graphs uniformly at random, meaning that each has probability $1/3$. In this example, the task is easy.

But what if there are thousands of vertices with an arbitrary prescribed degree vector? The number of possible graphs may be enormous, making it impossible to list them all and select one uniformly.

A natural solution is to use a random process called the *switch chain*.

The basic operation is remarkably simple. Suppose that four distinct vertices a, b, c, d satisfy the following conditions: ab and cd are edges, while ac and bd are not. We may then remove the edges ab, cd and insert the edges ac, bd.

Schematically,

$$\{ab, cd\} \longrightarrow \{ac, bd\}.$$

Such an operation is called a *switch*. It preserves the degree of every vertex, since each of the four vertices loses one edge and gains one edge.

Starting from any graph with the prescribed degree vector, we can repeatedly perform randomly selected switches. Since the degrees never change, we remain within the collection of graphs that we want to sample.

The hope is that after sufficiently many random switches, the resulting graph becomes almost uniformly distributed among all graphs with the given degrees.

More precisely, let $\Omega_{\mathbf{d}}$ denote the collection of all simple graphs on the labelled vertices $\{1, \ldots, n\}$ having degree vector $\mathbf{d}$.

We use the following version of the switch chain. At each step, with probability $1/2$, we do nothing. Otherwise, we select four distinct vertices uniformly at random and choose uniformly one of the six ordered pairs (F, F') of different ways of pairing these four vertices into two disjoint edges. If both edges of F are present and both edges of F' are absent, we replace F by F'. Otherwise, we do nothing.

The additional probability of doing nothing makes the process *lazy*, preventing certain periodic behaviour.

The process is an example of a *Markov chain*: the probabilities of its next state depend only on its current state, not on how it arrived there.

Moreover, every switch can be reversed with exactly the same probability. Together with the fact that any two graphs with the same degree vector can be connected by a sequence of switches, this ensures that the uniform distribution is the limiting distribution.

The difficult question is *how quickly* this convergence takes place.

To measure the difference between two probability distributions μ and ν on a finite set Ω, we use their *total variation distance*,

$$\|\mu - \nu\|_{\mathrm{TV}} = \frac{1}{2} \sum_{G \in \Omega} |\mu(G) - \nu(G)|.$$

Equivalently, this is the largest possible difference between the probabilities assigned by μ and ν to the same event.

If this distance is small, the two distributions are statistically difficult to distinguish.

The conjecture, associated with the work of Kannan, Tetali, and Vempala (Kannan et al. 1999), predicted that the switch chain approaches the uniform distribution in a number of steps bounded by a polynomial in n, for every graphical degree vector.

Substantial progress was made over the following decades. In 2007, Cooper, Dyer, and Greenhill (Cooper et al. 2007) proved polynomial mixing for regular graphs, in which every vertex has the same degree. Subsequent results covered various classes of irregular degree vectors.

In 2026, Fu, Qin, and Wang (Fu et al. 2026) established the corresponding result for bipartite graphs, where the vertices are divided into two classes and all edges join vertices from different classes.

However, the general problem for simple undirected graphs with arbitrary degrees remained open.

The new OpenAI preprint (OpenAI 2026) claims to settle this problem, and gives an explicit polynomial bound.

Theorem 1 (OpenAI) *Let $\mathbf{d}$ be any graphical degree vector on $n \geq 4$ labelled vertices. Starting from any graph in $\Omega_{\mathbf{d}}$, the lazy switch chain described above produces, after $2n^8$ steps, a probability distribution whose total variation distance from the uniform distribution on $\Omega_{\mathbf{d}}$ is at most $1/4$.*
Moreover, for every finite graphical degree vector, there exists a randomized algorithm that produces a graph exactly uniformly *from $\Omega_{\mathbf{d}}$. The algorithm terminates with probability one, and its expected running time is polynomial in the length of the input.*

The striking feature of the theorem is that it imposes *no restrictions whatsoever* on the degree vector, beyond the obvious requirement that at least one graph with these degrees exists.

The degrees may be very different from one another, the graph may be sparse or dense, and some vertices may have degrees close to 0 while others have degrees close to $n - 1$.

In every case, the same upper bound of $2n^8$ steps applies.

The bound is certainly not expected to be optimal, but its significance is that it is polynomial and completely uniform.

The second assertion is also worth emphasizing. Running a rapidly mixing Markov chain for a finite number of steps gives a distribution that is *approximately* uniform, not exactly uniform.

The theorem guarantees something stronger: a separate algorithm that produces each graph in $\Omega_{\mathbf{d}}$ with precisely the same probability.

Let us briefly discuss the proof.

The authors first introduce a more powerful operation than an ordinary switch. Instead of changing two edges, they choose a pair of vertices and randomly redistribute their neighbours, while preserving all vertex degrees.

Such an operation can change many edges simultaneously, but its possible outcomes have a particularly convenient mathematical structure.

The central part of the proof establishes inequalities controlling how quickly these larger random updates eliminate differences between probability distributions. The argument uses variance estimates and linear algebra, with special attention to the interactions between updates involving overlapping or disjoint pairs of vertices.

The authors then compare these larger updates with ordinary switches. This gives a quantitative bound on the *spectral gap* of the switch chain, a number measuring how quickly the process approaches equilibrium.

The resulting estimates imply the explicit mixing-time bound $2n^8$.

To obtain an exactly uniform sampler, the authors use an additional correction procedure. Most of the time, the algorithm simply runs the switch chain long enough to produce an extremely accurate approximation to the uniform distribution.

On a very rare branch, it performs a more expensive computation to correct the remaining bias exactly. Although this exceptional computation may take exponential time, it occurs with such small probability that the *expected* running time remains polynomial.

Thus, exact uniform sampling is possible in expected polynomial time, even though the algorithm does not have a polynomial worst-case running-time guarantee.

This result connects naturally with our earlier discussion of random regular graphs, where we studied the expansion properties of graphs selected at random. Here the focus is different: rather than analyzing a random graph after it has been generated, we ask how to generate it efficiently while preserving its prescribed degrees.

Uniform sampling with fixed degrees is particularly useful in network science, where one often wants to distinguish structural features caused merely by vertex degrees from genuinely unexpected properties of a network.

The result is currently available as a preprint. Its main conclusion is remarkably general: no matter how irregular the degree vector is, repeated random switches provide a polynomial-time route to an almost uniformly random graph, and an additional procedure makes exact uniform sampling possible.

References

Cooper, Colin, Martin Dyer, and Catherine Greenhill. 2007. “Sampling Regular Graphs and a Peer-to-Peer Network.” *Combinatorics, Probability and Computing* 16 (4): 557–93. https://doi.org/10.1017/S0963548306007978.

Fu, W., Q. Qin, and G. Wang. 2026. *Spectral Gap for the Binary Fixed-Margin Swap Chain*. https://arxiv.org/abs/2606.22636.

Kannan, Ravi, Prasad Tetali, and Santosh Vempala. 1999. “Simple Markov-Chain Algorithms for Generating Bipartite Graphs and Tournaments.” *Random Structures & Algorithms* 14 (4): 293–308.

OpenAI. 2026. *Polynomial Mixing of the Switch Chain for Every Graphical Degree Sequence*. OpenAI Math Release preprint. https://github.com/openai/math/blob/main/preprints/Polynomial-Mixing-of-the-Switch-Chain-for-Every-Graphical-Degree-Sequence-September-25-2026/main.pdf.