Random Generation of Directed Acyclic Graphs

G. Melançon [a], I. Dutour [b] and M. Bousquet-Mélou [b]

[a] *LIRMM, UMR CNRS 5506, France*
`Guy.Melancon@lirmm.fr`

[b] *LaBRI, UMR CNRS 5800, France*
`{dutour,bousquet}@labri.fr`

Abstract

We propose a simple algorithm based on a Markov chain for generating acyclic digraphs with a given number of vertices *uniformly at random*. In addition, standard combinatorial techniques allow us to describe the overall shape and average edge density of an acyclic digraph. As an application, we were able to estimate the density function of classical statistics on acyclic digraphs.

1 Introduction

This paper [1] describes a method for the generation of random acyclic directed graphs (digraphs). Our study of acyclic digraphs has been motivated by their importance as basic data structures for a number of Information Visualization applications. Acyclic digraphs have been at the heart of many problems in Graph Drawing and are still the subject of active research [2]. Because many of the algorithmic problems on acyclic digraphs are NP-complete or NP-hard, the literature on Graph Drawing or Information Visualization addresses significant issues which rely on "soft" argumentations. For instance, layout algorithms are often judged on aesthetic results, the expected statistical characteristics of metrics are used to visualize graphs and/or to control interaction, measurements on the average running time is often used to make a proper choice among layout algorithms, etc. To be more solid, all these argumentations require experimentation on a test suite consisting of a large

[1] An extended and slightly different version of the paper appeared as a technical report while the first author was still affiliated with the CWI institute. See `www.cwi.nl/InfoVisu`

number of randomly selected graphs. This motivates the accessibility of a reliable method to generate those at will.

We give an algorithm based on a Markov chain to generate uniformly distributed random acyclic digraphs of a given size. Fortunately, the enumeration of acyclic digraphs is rich enough to enable us to predict some of the properties the acyclic digraphs have on average. Furthermore, our generation algorithm can deal with various constraints and can be adapted to produce uniformly distributed random acyclic digraphs with bounded total degree, or bounded vertex degree. Hence our method is flexible and offers a way of controlling edge density of the resulting graphs, which makes it particularly valuable to researchers in Information Visualization. Graph Visualization often makes use of statistics on graphs. Using our algorithm we were able to gather empirical knowledge and conjecture analytical functions describing the (probability) density functions of classical statistics on acyclic digraphs.

2 A Markov chain algorithm

The exhaustive algorithm of listing all possible acyclic digraphs and choosing one at random is certainly not viable, since the number a_n of acyclic digraphs on n vertices grows too rapidly (the value $\log a_n$ grows as a quadratic function of n). The algorithm we propose relies on a Markov chain process for building an acyclic digraph with a prescribed number of vertices uniformly at random, starting from the empty graph. Our method is inspired from the work by Denise *et al.* [4] for generating random planar graphs.

Let $V = \{1, \ldots, n\}$ denote the set of underlying vertices of the graphs we consider. We define a Markov chain M with state space all acyclic digraphs over the set of vertices V. A Markov chain is completely determined by its transition function prescribing the probability that the chain goes from a given state to any other possible state. In our case, the transition function is as follows. Let us agree that a *position* consists of an ordered pair (i, j) of distinct vertices of V. If X_t denotes the state of the Markov chain at time t, then X_{t+1} is chosen according to the rules (a) and (b) below. Suppose a position (i, j) is chosen uniformly at random.

(a) If the position (i, j) corresponds to an arc e in X_t, then $X_{t+1} = X_t \setminus e$. That is, the edge e is deleted from the graph associated with X_t.
(b) If the position (i, j) does not correspond to an arc in X_t, then X_{t+1} is obtained from X_t by adding this arc, provided that the underlying graph remains acyclic; otherwise $X_{t+1} = X_t$.

It is easy to verify that (X_t) is an irreducible aperiodic Markov chain whose transition matrix is symmetric. Thus, (X_t) has a limiting stationary distribution which is uniform over the set of all acyclic digraphs over the set of vertices V. Indeed, M being *ergodic*, it converges to a unique stationary distribution, and one easily checks that the uniform distribution is indeed stationary (see, e.g., [3,8]). The stationary distribution is reached whatever initial distribution we take. So in particular, starting the process on the empty graph gives an effective way of generating an acyclic digraph over the set V uniformly at random. As is always the case, the closeness of the approximation is governed by the mixing rate of the chain (and the choice of the initial distribution). The problem of finding how long the chain has to be iterated in order to be ϵ-close to the uniform distribution is a difficult one. Observe that the maximal distance between any two acyclic digraphs is bounded by $n(n-1)$, since an obvious (but far from optimal) path connecting them goes through the empty graph (by first deleting all edges from the first graph and then adding the edges of the second graph). This fact is in accordance with our experimental observation that iterating the chain a quadratic number of times brings the process acceptably close to the uniform distribution.

Proposition 1 *Assuming that the Markov chain stabilizes after cn^2 steps, then the algorithm it provides for generating an acyclic digraph uniformly at random has worst case time complexity $O(n^4)$.*

At each step of the algorithm where an edge is potentially added (not already in the underlying graph), we must check if its addition creates a cycle. That check can be efficiently done, using a breath-first-search in the graph, and relying on a queue not using more than linear space adapted from the algorithm by Itai and Rodeh [1, Section 6.1]. Their algorithm runs in time $O(n + e)$ (cf [1, Th. 5]); since $e \leq n(n-1)/2$ (the complete graph admits an acyclic orientation), we get the result. Itai and Rodeh show that their cycle checking algorithm works in time $O(n \log n)$ on average. However, this result cannot be straightforwardly used here and it is unclear how to improve the theoretical estimation of the complexity even though the complexity of our algorithm seems to be clearly below $O(n^4)$.

2.1 Properties of random acyclic digraphs

Random generation is often used as an experimental approach to discover properties of the generated objects. However, the generating function for the number of acyclic digraphs can be used to get explicit asymptotic values for various parameters, as confirmed by the following statement (see the extended version for details).

Theorem 2 *The average number of edges in an acyclic digraph on n vertices* *is* $\sim n^2/4$.

The number of out-points (source points with no predecessors) of an acyclic digraph could be interpreted as a measure of its *width*. At least when dealing with random acyclic digraphs, our experiments show that we may expect this measure to be uniform through the levels of the graphs and thus this interpretation makes sense. A striking fact is that when computing the average number of out-points in acyclic digraphs we find that:

$$\lim_{n\to\infty} \frac{\sum_{k=1}^{n} k a_n^{(k)}}{a_n} = \rho \sim 1.4880785456 \tag{1}$$

where ρ is the first (and simple) zero of a function related with the generating function of acyclic digraphs according to their number of out-points ($a_n^{(k)}$ denotes the number of acyclic digraphs on n vertices and k out-points). Incidentally, the methods used by Rodionov [7] can be adapted to deal with the generating series of acyclic digraphs according to their number of vertices and number of out-points [2]. Gessel [5] gave an identity from which we were able to prove Eq. (1).

3 Applications

3.1 Restricting the state space to gain control over edge density

The properties of a random acyclic digraph as given above (Th. 2) describe its overall shape and also indicates that it has a very high edge density. This density is much higher than what is usually considered as "dense" by the Information Visualization community (a graph with n nodes and $2n$ edges is sometimes considered dense). However, we shall describe here how the Markov chain can be slightly modified to work on a state space consisting of graphs with lower edge density, thus providing generators for random acyclic digraphs that fit the needs of Information Visualization and Graph Drawing.

Imposing on step (b) of the Markov chain the additional condition that, for example, the total degree of a graph remains below a given bound d does not violate any of the required property for the chain to be ergodic. Indeed, the state space remains irreducible and the transition function preserves symmetry and aperiodicity. Thus iterating the process with this additional requirement provides a way of generating acyclic digraphs with prescribed edge density

[2] The extended version lists previous works by Robinson, Liskovets and others giving asymptotic estimations of various parameters on acyclic digraphs

uniformly at random. Since the Markov chain works by systematically adding an edge whenever possible, it will indeed produce graphs with a number of edges very close to the fixed upper bound. It should clearly be understood here that this process randomly generates acyclic digraphs *from* the set of all acyclic digraphs (over a given number of vertices) not having more than d edges. Similar restrictions, such as bounded vertex degree using the same bound for all vertex or using a specific bound for each vertex can also be applied (see the extended version for details).

3.2 Classical statistics on acyclic digraphs

Information Visualization makes use of various statistics on graphs to produce views that can be easily navigated [6]. One familiar statistic on acyclic digraphs, we call the flow statitic, can be defined in analogy to Kirchoff's law on electrical circuits. Input values are assigned to source nodes; a node that has been assigned a value broadcasts it to its successors after dividing it by its outdegree. The value of a (non source) node is obtained by adding values coming from its predecessors. In [6], it is shown that knowledge of the distribution of values in an acyclic digraph is necessary is various situations to provide a faithful image of the data (when assigning colors to edges, for example). However, the discrete nature of our computations makes it necessary to work with the density function (or cumulative probability) of the statistic instead of working with the distribution itself [3]. Using our algorithm we were able to experimentally compute the flow statistic on a large sample of graphs and conjecture analytical functions approaching its density.

Conjecture 3 *Let $\mathcal{A}_d$ denote the class of all acyclic digraphs having edge density d (the edge density of a graph is $d = e/n$). The density function f_d for the flow statistic on acyclic digraphs with edge density equal to d is given by $f(x) = 1 - Ke^c$, where K and c are constants depending on the edge density d.*

4 Conclusion and future work

Proposition 1 is far from being satisfactory from a theoretical standpoint. A deeper study of the Markov chain itself, together with a better analysis of the cycle checking algorithm should reveal that our algorithm has a time complexity that is significantly below the above estimation.

[3] Most often, the density function f associated with a probability distribution P is $f(x) = \int_{-\infty}^{x} P(t)dt$.

The use of statistics on acyclic digraphs in Information Visualization led us to other conjectures similar to Conjecture 3. It is interesting to note that the problem here is not to evaluate the distribution of a parameter over the set of all acyclic digraphs (or acyclic digraphs of bounded total degree), but rather to try to have a good approximation of the distribution of values of a given statitic over the nodes or edges in a given acyclic digraph. Analytical approximations of the density functions then allow easy and efficient implementations of these functions and can be used in various applications.

References

[1] Alon, I. and Rodeh, M. Finding a Minimal Circuit in a Graph. *SIAM Journal of Computing*, 7(4):413–423, 1978.

[2] Battista, G.d., Eades, P., Tamassia, R., and Tollis, I.G. *Graph Drawing: Algorithms for the Visualisation of Graphs.* Prentice Hall, 1999.

[3] Berman, A. and Plemmons, R. J. *Non-negative matrices in the mathematical sciences.* Computer Science and Applied Mathematics. Academic Press, London, 1979.

[4] Denise, A., Vasconcellos, M., and Welsh, D.J.A. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.

[5] I. Gessel. Counting acyclic digraphs by sources and sinks. *Discrete Mathematics*, 160:253–258, 1996.

[6] Herman I., Marshall M. S., and Melançon G. Density functions for visual attributes and effective partitioning in graph visualization. In *IEEE Symposium on Information Visualization (InfoVis'2000)*, pages 49–56, Salt Lake City, Utah, U.S., 2000. IEEE Computer Society.

[7] V. I. Rodionov. On the number of labeled acyclic digraphs. *Discrete Mathematics*, 105:319–321, 1992.

[8] A. Sinclair. *Algorithms for Random Generation & Counting: A Markov Chain Approach.* Progress in Theoretical Computer Science. Birkhaüser, Boston, 1993.