Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Random Generation of Dags for Graph Drawing

G. Melancon, I. Dutour, M. Bousquet-Melou

Information Systems (INS)

INS-R0005 February 29, 2000

Report INS-R0005
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Random Generation of Dags for Graph Drawing

G. Melançon
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Guy.Melancon@cwi.nl

I. Dutour, M. Bousquet-Mélou
LaBRI, UMR 5800
351, Cours de la Libération
33405 Talence Cedex, France
{dutour, bousquet}@labri.u-bordeaux.fr

ABSTRACT
The Graph Drawing community uses test suites for comparing layout quality and efficiency. Those suites often claim to collect randomly generated graphs, although in most cases randomness is a loosely defined notion. We propose a simple algorithm for generating acyclic digraphs with a given number of vertices *uniformly at random*. Applying standard combinatorial techniques, we describe the overall shape and average edge density of an acyclic digraph. The usefulness of our algorithm resides in the possibility of controlling edge density of the generated graphs. We have used our technique to build a large test suite of acyclic digraphs with various edge density and number of vertices ranging from 10 to 1000.

2000 Mathematics Subject Classification: 05A15, 05A16, 05C80, 68U05
Keywords and Phrases: graph drawing, information visualization, directed acyclic graphs, random generation
Note: At CWI, the work was carried under the project INS3.2 “Information Visualization”.
The on-line version of this report contains parts of the figures in colour[1].

1. Introduction

Although a large number of results which are published in the area of Graph Drawing are based on deterministic mathematical facts, there are also other significant issues which rely on “softer” argumentations. Layout algorithms are often judged on aesthetic results [5]; the expected statistical characteristics of metrics are used to visualize graphs and/or to control interaction [22, 12, 14]; 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 number of graphs. Moreover, to be statistically reliable, the underlying distribution of the test suite should be known, and the sample generation process should be reproducible. Hence the importance for the Graph Drawing community to have an access to random sequences of graphs and, moreover, to have a reliable method to generate those at will.

In specific cases, working with a general uniform distribution of graph samples might not be satisfactory. It is often necessary to build a test suite constrained by some specific characteristics. As an example, one might want to test a specific layout algorithm on a suite of graphs whose average edge density fall within some well defined bounds, or where the degree for each node is below than a specific value. When applying Graph Drawing to Information Visualization, for example, the characteristics

[1] See `ftp://ftp.cwi.nl/pub/CWIreports/INS/INS-R0005.ps.Z`

n	1	2	3	4	5	6	7	8	9
a_n	1	3	25	543	29281	3781503	1138779265	783702329343	1213442454842881

Table 1: Number of labeled acyclic digraphs on $n = 1, \ldots, 9$ vertices

of a layout algorithm on such samples might be of a primary importance in choosing the right layout algorithm [18]. Some suites have been recently proposed and have become used by the researchers in the field; they appear as a common point of reference for comparing efficiency and quality of new layout algorithms[2]. Most of these suites claim to be random, although this notion seems to be loosely defined. The fact that it is impossible to predict the outcome of a generation algorithm has no direct implication on the underlying distribution it follows, and certainly does not help to establish any result concerning it.

This paper describes a method for the generation of random suites of acyclic directed graphs (digraphs). The choice 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 [5].

When randomly selecting objects in a set, one usually relies on a distribution which describes the probability for each object to be selected. Most often, this distribution will be uniform, that is every object will be assigned the same probability, which is then completely determined by the number of objects in the set. However, when designing an ad hoc generating process, it might be difficult to prove that the underlying distribution is indeed uniform over the set of graphs it generates. An alternative approach is to start with a process for which the underlying distribution is known and then study the properties that the generated objects would share. It is this approach that we have followed. We give an algorithm, based on a Markov chain, to generate uniformly distributed random acyclic digraphs of a given size (Section 2). Fortunately, the enumeration of acyclic digraphs is rich enough to enable us to predict some of the properties the acyclic digraphs have on average. For example, we can compute the expected average number of edges as well as the expected number of out-points of an acyclic digraph with a given number of vertices (Section 2.2). Furthermore, our generation algorithm can deal with various constraints. In particular, we show how the Markov chain 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 for Graph Drawing research.

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; cf Table 1 below). 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.* [9] 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

[2]Some of those suites are available via the web, see for example:
`http://www.dia.uniroma3.it/~patrigna/3dcube/`
`http://www.research.att.com/sw/tools/graphviz/refs.html`.

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., [8, 2]). 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. Our experimentations tend to show that iterating the chain a quadratic number of times brings the process acceptably close to the uniform distribution (at least as close as needed when generating a test set of graphs for a layout). This generating process (with the same iterating bound) was used to build samples of graphs for comparing various implementations of layout algorithms [18]. Figure 1 gives an example of a graph with 20 vertices generated using the Markov chain.

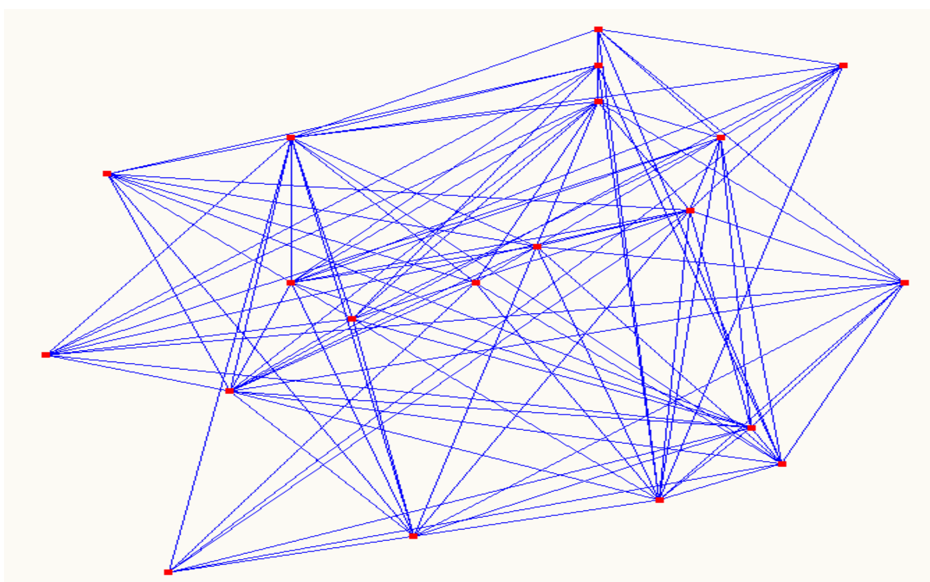

Figure 1: A randomly generated acyclic digraph with 20 vertices. The nodes are vertically positioned by layers.

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 works in average time $O(n^3 \log n)$.*

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 [3, Section 6.1]. Since the algorithm by Itai and Rodeh runs in time $n \log n$ on the average we get the result (cf [3, Th. 5]).

2.1 An alternative generating process

A "constructive" and alternative approach to the Markov chain process is possible. Robinson[20] gave a closed formula to compute the number of labeled acyclic digraphs that induces a decomposition process. Indeed, let G be an acyclic digraph with k *out-points* (points with incoming degree 0, usually drawn at the top of the diagram and sometimes called source points [15]); denote by K the set of these out-points in G (so $|K| = k$). Consider the graph G' whose vertices are $V' = V \setminus K$, with edges those between vertices of V' in G and denote by S the set of its out-points, with $|S| = s$. The graph G' is itself an acyclic digraph. Moreover in G, every vertex in S must be the successor of at least one vertex in K. Finally, apart from these arcs and those in G', there are arcs going from vertices in K to vertices in $V' \setminus S$. From which we get the recurrence [20]:

$$a_n^{(k)} = \sum_{s=1}^{n-k} \binom{n}{k} (2^k - 1)^s 2^{k(n-k-s)} a_{n-k}^{(s)}, \tag{2.1}$$

where $a_n^{(k)}$ denotes the number of acyclic digraphs with n vertices, k of which are out-points (with $a_n^{(0)} = 0$ if $k > 0$ and $a_0^{(0)} = 1$). This decomposition for acyclic digraphs could in principle be used to generate an acyclic digraph, using the following algorithm. Let $n \geq 0$ and $0 \leq k \leq n$ respectively denote the number of vertices and out-points of the acyclic digraph to generate.

1. Randomly select a non-empty subset K of $V = \{1, \ldots, n\}$ with k elements.
2. Randomly select an integer s between 1 and $n - k$ with probability distribution
$$P(s) = \binom{n}{k} (2^{k-1})^s 2^{k(n-k-s)} a_{n-k}^{(s)} / a_n^k.$$
Let us agree to denote by S a subset of s elements, chosen among $V \setminus K$. This set shall only be determined when recursively calling the procedure and applying the preceding step, but being able to already refer to it makes things clearer.
3. Randomly and independently select s *non-empty* subsets of K (one for each element in S).
4. Randomly and independently select $(n-k-s)$ *possibly empty* subsets of K (one for each element in $V \setminus (K \cup S)$).
5. Randomly select an acyclic digraph on $n - k$ vertices, having s out-points (it is at this stage that the set S shall be determined).
6. Use the various preceding components to build the graph G to output.

Note that this algorithm would first need to choose the integer k using the probability distribution $a_n^{(k)}/a_n$ ($k = 1, \ldots, n$). The implementation of such a procedure necessitates the use of the numbers $a_n^{(k)}$ for all n and k, which should be pre-calculated. This is possible, to some extent, since symbolic computation packages can easily compute $a_n^{(k)}$ for values of n and k up to a little less than a thousand. However, those packages are usually not efficient when performing demanding algorithms on graphs and in the end, a solution is to store the numbers in tables and implement the algorithm in a traditionnal programming environment. This solution still requires to manipulate the numbers $a_n^{(k)}$ and a_n. The fast growth of those numbers requires quadratic space and consequently, it is necessary to rely on a specialized library capable of dealing with the arithmetic of large numbers (see [19], for instance). Moreover, it is not at all clear that the above algorithm would have a complexity clearly below that given by the Markov chain.

2.2 Properties of random acyclic digraphs
Random generation is often used as an experimental approach to discover properties of the generated objects. In our case, as we shall see, experimentation is not needed since the generating function for the number of acyclic digraphs can be used to get explicit asymptotic values for various parameters. Although technical, this subsection illustrates how combinatorial analysis can help to describe the overall "aesthetic" of the set of acyclic digraphs. This is important when evaluating what cases the graphs output by the Markov chain cover (within the set of cases an algorithm should be tested on).

Average number of edges. Our first observation was that we could borrow a result from Denise *et al.* [9] to give a lower bound for the average number of edges in a random acyclic digraph.

Proposition 2 *The average number of edges in the random acyclic digraph over n vertices is at least $\frac{n(n-1)}{4}$.*

As in [9], this follows from a more general result we need to state in order to sketch a proof for Prop. 2. The proof is interesting in that it partly reveals the structure of the state space of the Markov chain we use to generate the acyclic digraphs.

Lemma 1 (cf [9, Theorem 2]) *Let E be a finite set, and $\mathcal{D}$ a family of subsets of E such that:*

(i) if X is in $\mathcal{D}$, and if $Y \subset X$ then Y is in $\mathcal{D}$;

(ii) all maximal members of $\mathcal{D}$ have the same cardinality m.

Then the expected number of elements of a random member of $\mathcal{D}$ is at least $\frac{m}{2}$.

Any acyclic digraph is uniquely determined by its set of arcs. Thus, we may apply Lemma 1 by taking for $\mathcal{D}$ the family of those sets of arcs of all possible acyclic digraphs on $V = \{1, \ldots, n\}$. Condition (i) is straightforward since any subgraph of an acyclic digraph is itself acyclic. The proposition then follows from the following claim: all maximal members in the family of (sets of arcs of) acyclic digraphs over a set of n elements have $n(n-1)/2$ arcs. To prove the claim we first note that any acyclic digraph has at most $n(n-1)/2$ elements. Indeed, of the two arcs (i,j) or (j,i) an acyclic digraph may contain at most one. Hence there is a well defined and unique undirected graph associated with any acyclic digraph. Moreover, the (undirected) complete graph can be given an acyclic orientation; this is easily achieved by defining an orientation for all edges using the rule: $\{i,j\}$ is oriented as (i,j) if and only if $i < j$. Hence, there are acyclic digraphs with $n(n-1)/2$ edges. Finally, observe that any acyclic digraph G with less than $n(n-1)/2$ edges can be extended to a graph having one more arc. Indeed, in that case there must exist a pair $\{i,j\}$ of non-adjacent vertices. Then either the arc (i,j) or the arc (j,i) can be added to G without introducing a directed cycle. If it was not the case, then we would conclude that there already is a cycle in G, which is impossible.

Actually, this lower bound is very close to the asymptotic value for the number of edges in a random acyclic digraph, as shows the following theorem:

Theorem 1 *The average number of edges in an acyclic digraph on n vertices is $\sim n^2/4$.*

Denote by $a_{n,p}$ the number of acyclic digraphs on n vertices with p edges. Consider the polynomial $A_n(x) = \sum_{p\geq 0} a_{n,p}x^p$ and the *generating function* $A(x,t) = \sum_{n\geq 0} \frac{A_n(x)}{n!} \frac{t^n}{(1+x)^{\binom{n}{2}}}$. Rodionov [16] proved the identity:

$$A(x,t) = (D(x,t))^{-1}, \quad \text{where } D(x,t) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!} \frac{t^m}{(1+x)^{\binom{m}{2}}}.$$

We need to evaluate the quantity $A'_n(1)/A_n(1)$. Differentiating $A(x,t)$ with respect to x and evaluating at $x = 1$, we get

$$\frac{\partial A(x,t)}{\partial x}\Big|_{x=1} = \sum_{n=0}^{\infty}\frac{A'_n(1)}{n!}\frac{t^n}{2^{\binom{n}{2}}} - \sum_{n=0}^{\infty}\frac{A_n(1)}{n!}\frac{t^n}{2^{\binom{n}{2}+1}}\binom{n}{2} \tag{2.2}$$
$$= -\frac{\partial D(x,t)}{\partial x}/D(x,t)^2\Big|_{x=1}$$

We write $D(t)$ for the value of $D(x,t)$ at $x=1$. Now, rearranging the second term in Eq. (2.2) we compute:

$$\sum_{n=0}^{\infty}\frac{A'_n(1)}{n!}\frac{t^n}{2^{\binom{n}{2}}} = \frac{t^2}{4}\frac{\partial^2}{\partial t^2}\frac{1}{D(t)} - \frac{\partial D(x,t)}{\partial x}/D(x,t)^2\Big|_{x=1}.$$

Combined with the easily obtained identity $\frac{\partial D}{\partial x}\big|_{x=1} = -\frac{t^2}{4}\frac{\partial^2}{\partial t^2}D(t)$, we get:

$$\sum_{n=0}^{\infty}\frac{A'_n(1)}{n!}\frac{t^n}{2^{\binom{n}{2}}} = \frac{t^2}{4}\left(\frac{\partial^2}{\partial t^2}\frac{1}{D(t)} + \frac{\frac{\partial^2}{\partial t^2}D(t)}{D(t)^2}\right) = \frac{t^2}{2}\frac{(\frac{\partial D(t)}{\partial t})^2}{D(t)^3} \tag{2.3}$$

We conclude from the previous calculations, that our approximation will depend on our knowledge of the function $D(t)$, and more particularly on the knowledge of its zeroes and its singularities. Now, it turns out that this function is entire and that all its zeros are real. This has already been observed by Robinson [20]; he gave a value for the first zero of $D(t)$, which is $\rho \sim 1.4880785456$. Moreover, this zero is *simple*. Expanding $D(t)$ around ρ and substituting this into Eq. (2.3), we obtain:

$$\frac{t^2}{2}\frac{(\frac{\partial D(t)}{\partial t})^2}{D(t)^3} \simeq \frac{\rho^2}{2}\frac{D'(\rho)^2}{(t-\rho)^3 D'(\rho)^3},$$

where $D'(\rho)$ denotes $\frac{\partial D}{\partial t}(t)$ evaluated at $t=\rho$. This in turn is equal to $-\frac{1}{2\rho D'(\rho)(1-t/\rho)^3}$. Expanding the term $(1-t/\rho)^3$ to get the coefficient of t^n, we find:

$$\frac{A'_n(1)}{n!}2^{-\binom{n}{2}} \sim -\frac{1}{2\rho D'(\rho)}\frac{1}{\rho^n}\frac{n^2}{2} = -\frac{n^2}{4\rho^{n+1}D'(\rho)}.$$

The predicted approximation is a consequence of $\frac{A_n(1)}{n!}2^{-\binom{n}{2}} \sim -1/(\rho^{n+1}D'(\rho))$, which follows from $A(1,t) \sim -1/(\rho D'(\rho)(1-t/\rho))$, by expanding $D(1,t)$ around ρ. Observe that Theorem 1 implies that the average degree of a node is $n/2$.

Remark 1 Robinson was the first to enumerate acyclic digraphs. Bender *et al.* [7] gave asymptotic formulas for the numbers $a_{n,p}$. Robinson also looked at the problem of enumerating *unlabeled* acyclic digraphs [21]. Randomly generating unlabeled acyclic digraphs would perfectly fit the needs of Graph Drawing, since the labels serve no particular purpose here; moreover, the number of unlabeled structures are way below those stated in Table 1 (see Table 2 below). However, this is in general a more difficult problem since it involves dealing with the symmetry group of the underlying structures. See [21, 6] for results on those numbers and their asymptotic approximations.

n	1	2	3	4	5	6	7	8	9
a_n	1	2	6	31	302	5984	243668	20286025	3424938010

Table 2: Number of unlabeled acyclic digraphs on $n = 1, \dots, 9$ vertices

Width and height of acyclic digraphs. The number of out-points (source points) of an acyclic digraphs 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. The problem of counting the number of acyclic digraphs with respect to their numbers of vertices and number of out-points was addressed by Liskovec [1] and Gessel [15]. A striking fact is that when computing the average number of out-points in acyclic digraphs we find that:

$$\lim_{n\to\infty} \frac{\sum_{k=1}^{n} k a_n^{(k)}}{a_n} = \rho \tag{2.4}$$

where ρ is the first (and simple) zero of the function $D(t)$ defined above. Incidentally, the methods used by Rodionov [16] can be adapted to deal with the generating series of acyclic digraphs according to their number of vertices and number of out-points. Gessel [15] gave an identity from which we were able to prove Eq. (2.4).

Similarly, the *height* of an acyclic digraph can be defined as its number of levels, that is the height of the graph is equal to the length of a maximal path starting from an out-point in the graph. Our experimentations indicate that this number is about $2n/3$ which seems coherent with the intuition that the product of the height and the width of an acyclic digraph should be n.

3. Gaining control over edge density

The properties of a random acyclic digraph as given above 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" (a graph with n nodes and $2n$ edges is sometimes considered dense). Our generation process cannot be used to produce acyclic digraphs with edge density higher than $n^2/4$, but this can certainly not be seen as a defect (!). 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 Graph Drawing.

Imposing on step (b) of the Markov chain the additional condition that, for example, the total degree of a graph does not exceed 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 bounded edge density uniformly at random (within the set of acyclic digraphs having total degree $\leq d$). 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.

Similarly, we could impose a condition on the degree of vertices, asking that it does not exceed a given bound d. Again, the chain remains ergodic and we get a generating process offering control over edge density. In this case, the edges will be distributed over the vertices in a more uniform way than with the preceding condition. With this particular condition, the Markov chain will output graphs with a number of edges close to $nd/2$ (since every vertex will tend to have a number of edges close to d). Figure 2 gives an example of a graph on 30 vertices generated with the Markov chain with the condition that the vertex degree does not exceed degree $d = 4$.

Observe that the results given in section 2.2 describing the overall shape of a random acyclic digraph are not valid anymore. For instance, acyclic digraphs obtained when using for d a value as small as $d = 2$ will produce acyclic digraphs with many out-points. We have used our technique to build a large test suite[3] of acyclic digraphs with various edge density and number of vertices ranging from 10 to 1000. Our algorithm can also be adapted to produce acyclic digraphs with varying edge density, by first generating an acyclic digraph using the Markov chain (with a given upper bound on vertex

[3] See the site www.cwi.nl/InfoVisu

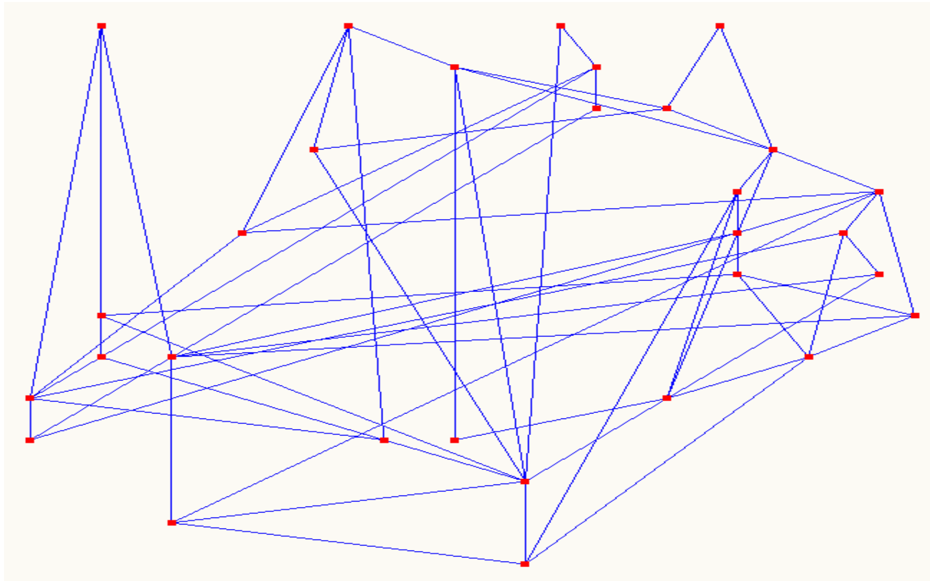

Figure 2: Randomly generated acyclic digraph with 30 vertices and edge density bounded by 60 (twice the number of nodes)

.

degree, say), and then select a subset of the nodes on which additional edges are added (using the Markov chain again).

4. Conclusion and future work

As mentioned earlier, our approach is very much inspired by the work of Denise *et al.* [9] on random planar graphs. Deterministic algorithms exist for randomly generating various families of trees [4]. Schaeffer [10, 11] recently proposed efficient procedures for random generation of planar maps. The problem of randomly generating structures is often closely related to that of enumerating them. Combinatorial mathematics can certainly help the Graph Drawing community by providing tools to help investigate "aesthetic" properties of certain families of graphs. Other families of graphs, which seem relevant for Graph Drawing, have not been looked at from an enumerative point of view. Leveled graphs, for example, form an important subclass of graphs studied by the Graph Drawing community for which the methods presented here do not directly apply.

Our graph generation methods have been implemented as a (command–line style) Java application, which generates its output either in GML or in a local vocabulary of XML used to describe graphs for Graph Visualization [13]. Our application consists of a small number of Java classes that implement the Markov chain and the cycle–checking algorithm by Itai and Rodeh [3]. The data structures we use to store and manipulate the graphs use a Java API called the Graph Visualization Framework [17], which is available on the web[3].

Acknowledgements. We wish to thank A. Denise for e-mail discussions at the early stage of this work. A. Raspaud also helped us in answering various questions on acyclic digraphs. We also thank I. Herman for comments on our manuscript.

References

1. Liskovec V. A. On the number of maximal vertices of a random acyclic digraph. *Theor. Probab. Appl.*, 20:401–409, 1975.
2. Sinclair A. *Algorithms for Random Generation & Counting: A Markov Chain Approach.* Progress in Theoretical Computer Science. Birkhaüser, Boston, 1993.
3. Alon I. and Rodeh M. Finding a Minimal Circuit in a Graph. *SIAM Journal of Computing*, 7(4):413–423, 1978.
4. Alonso. L, Rémy J.-R., and Schott. R. A linear-time algorithm for the generation of trees. *Algorithmica*, 17:162–182, 1997.
5. Battista G.d., Eades P., Tamassia R., and Tollis I.G. *Graph Drawing: Algorithms for the Visualisation of Graphs.* Prentice Hall, 1999.
6. Bender E. A., Richmond. L. B., and Robinson R. W. The asymptotic number of acyclic digraphs, ii. *Journal of Combinatorial Theory, Series B*, 44:363–369, 1988.
7. Bender E. A., Richmond. L. B., Robinson R. W., and Wormald N. C. The asymptotic number of acyclic digraphs, I. *Combinatorica*, 6(1):15–22, 1986.
8. Berman A. and Plemmons R. J. *Non-negative matrices in the mathematical sciences.* Computer Science and Applied Mathematics. Academic Press, London, 1979.
9. Denise A., Vasconcellos M., and Welsh D.J.A. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.
10. Schaeffer G. Bijective census and random generation of Eulerian planar maps. *Electronic Journal of Combinatorics*, 4(R20):14 pp., 1997.
11. Schaeffer G. Random sampling of large planar maps and convex polyhedra. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC '99)*, pages 760–769, Atlanta, Georgia, USA, 1999. ACM.
12. Herman I., Delest M., and Melançon G. Tree Visualisation and Navigation Clues for Information Visualisation. *Computer Graphics Forum*, 17(2):153–165, 1998.
13. Herman I. and Marshall M. S. GraphXML. Technical report, Centrum voor Wiskunde en Informatica (CWI), 2000. See `www.cwi.nl/InfoVisu`.
14. Herman I., Marshall S.M., Melançon G., Duke D., Delest M., and Domenger J.-P. Skeletal Images

as Visual Cues in Graph Visualization. In *Data Visualization '99*, pages 13–22, Vienna, Austria, 1999. Springer-Verlag.

15. Gessel I. Counting acyclic digraphs by sources and sinks. *Discrete Mathematics*, 160:253–258, 1996.
16. Rodionov V. I. On the number of labeled acyclic digraphs. *Discrete Mathematics*, 105:319–321, 1992.
17. Marshall M. S., Herman I., and Melançon G. An Object-Oriented Design for Graph Visualization. Technical Report INS-0001, Centrum voor Wiskunde en Informatica (CWI), 2000. See `ftp://ftp/cwi/nl/pub/CWIreports/INS`.
18. Melançon G. and Herman I. Dag drawing from an information visualization perspective. In van Liere R., editor, *Data Visualization '00*, Amsterdam, The Netherlands, to appear. Springer–Verlag.
19. Granlund T. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 2.0.2 edition, 1996. `http://www.swox.com/gmp/index.html`.
20. Robinson R. W. Counting labeled acyclic digraphs. In F. Harary, editor, *New Directions in the Theory of Graphs*, pages 28–43, Ann Arbor, Michigan, 1973. Academic Press.
21. Robinson R. W. Counting unlabeled acyclic digraphs. In C. H. C. Little, editor, *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, pages 239–273, Australia, 1977. Springer-Verlag.
22. Wilson R. and Bergeron R. Dynamic Hierarchy Specification and Visualization. In *Information Visualization '99*, pages 65–72, San Francisco, Califronia, USA, 1999. IEEE CS.