Rend. Istit. Mat. Univ. Trieste
Vol. 53 (2021), Art. No. 22, pp. 1–8 (electronic)
DOI: 10.13137/2464-8728/33308

Random generation of essential directed acyclic graphs

Romeo Rizzi and Alexandru I. Tomescu

Thank you Eugenio for your infinite encouragement, spiraling conversations and hyper-science, during the wonderful years 2009–2011 in Udine, Trieste and Monfalcone.

Abstract. *A directed acyclic graph (DAG) is called* essential *if for every edge (u, v) it holds that the set of in-neighbors of u is different than the set of in-neighbors of v minus vertex u. Essential DAGs have applications in Bayesian networks, where a basic problem is to generate uniformly at random a labeled essential DAGs with a given number of vertices. In this paper we prove a new decomposition of essential DAGs, which entails: (i) a new counting recurrence, and (ii) a new random generation algorithm, that may be of potential use for their applications in Bayesian networks.*

Keywords: Directed acyclic graph, counting recurrence, sampling.
MS Classification 2020: 05C20, 05C30, 68R10, 05C80.

1. Introduction

Directed acyclic graphs (*DAGs*) are a most basic class of graphs, with applications in various areas of Mathematics and Computer Science. For example, in Set Theory, hereditarily finite well-founded sets correspond to a subclass of DAGs (*extensional DAGs*) when one interprets each set as a vertex, and each membership relation $x \ni y$ as an edge (x, y), see e.g. [7]. In Computer Science, a DAG guarantees, e.g. the existence of a *topological order* among its vertices, enabling various optimal algorithms on DAGs, for problems that are otherwise NP-hard on general graphs (e.g. longest path).

A basic combinatorial question about DAGs is: given a natural number n, how many *labeled* DAGs with n vertices $\{1, \ldots, n\}$ exist? Here, by "labeled" we mean that we consider two labeled DAGs to be the same if and only if they have the same set of edges (see Figure 1 for an example). For general DAGs, the first counting results date back to [2, 3, 5, 12, 13]. This question has been addressed for other subclasses of DAGs, for example extensional DAGs [8, 10, 16].

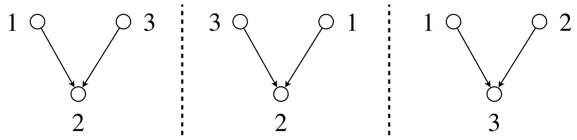

Figure 1: Three labeled DAGs with vertex set $\{1, 2, 3\}$. The two left-most DAGs are the same, since their set of edges is the same, namely $\{(1, 2), (3, 2)\}$. The DAG on the right is not the same as the other two ones, since its set of edges is $\{(1, 3), (2, 3)\}$.

In this paper we consider a different subclass of DAGs, namely *essential DAGs*. Given a DAG $G = (V, E)$, we denote its vertex set V also by $V(G)$. Given $v \in V(G)$, we denote by $N^+(v)$ the set of *out-neighbors* of v, namely $N^+(v) = \{w : (v, w) \in E\}$, and by $N^-(v)$ the set of *in-neighbors* of v, namely $N^-(v) = \{u : (u, v) \in E\}$. We say that G is *essential* if for any edge (u, v) it holds that $N^-(u) \neq N^-(v) \setminus \{u\}$. See Figures 2 and 3 for examples. Essential DAGs are used to represent the structure of Bayesian networks [1, 4, 9]. More specifically, [1] shows that essential DAGs are representatives of the Markov-equivalence classes of acyclic Bayesian networks (i.e., are Markov-equivalent to all acyclic networks in its equivalence class). By working only with such representatives, [1] claims that various computational problems on acyclic Bayesian networks are more efficient, or various constraints on prior distributions can be dropped in model selection algorithms.

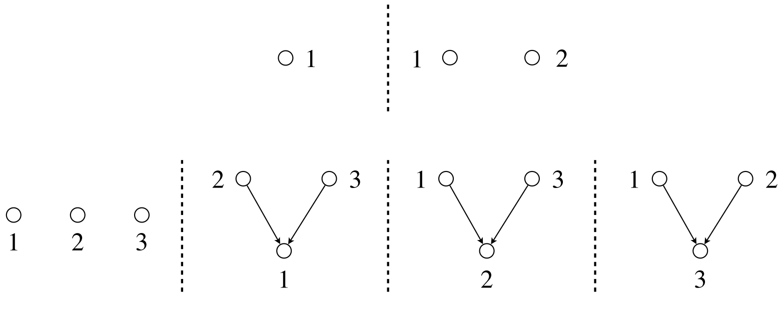

Figure 2: All the labeled essential DAGs with one vertex (top left), two vertices (top right), three vertices (bottom).

Essential DAGs were counted in [14] by inclusion-exclusion, and the asymptotic behavior of their number was studied in [15]. We should also note the similarity between the definition of essential DAGs and that of extensional

DAGs, the later of which requiring that for any two distinct vertices u and v, it holds that $N^+(u) \neq N^+(v)$ [7].

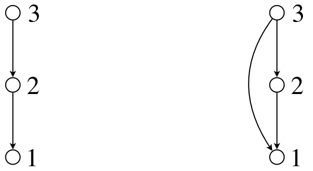

Figure 3: Two DAGs that are not essential. In the DAG on the left, for the edge $(3,2)$ it holds that $\emptyset = N^-(3) = N^-(2) \setminus \{3\}$. In the DAG on the right, for the edge $(2,1)$ it holds that $\{3\} = N^-(2) = N^-(1) \setminus \{2\}$.

Given their applications in Bayesian networks (e.g., in model selection, see [1]), a basic problem on essential DAGs is: given a natural number n, generate uniformly at random (*u.a.r.*) a labeled essential DAGs with n vertices. This problem can be solved by a result showing that the number of labeled DAGs with n vertices and the number of labeled essential DAGs on n vertices are asymptotically the same, up a factor of ≈ 13.65 [15]. This guarantees that one can generate u.a.r. an essential DAG by repeatedly generating u.a.r. a general DAG (using e.g. [6]) until getting one that is essential.

In this paper, we show another, direct approach, using the same recursive approach used for extensional DAGs from [7, Chapter 7]. For achieving this, we introduce a new decomposition of essential DAGs, and first show that it leads to a new counting recurrence for them. Then, we exploit this new insight in obtaining the direct recursive random generation procedure. This paper contains results from our preprint [11].

2. Counting

Our decompositions of essential DAGs is symmetric to the one by *rank* introduced in [10] for extensional DAGs, where the *rank* of a vertex v of a DAG is defined as the length of a longest path from v to a *sink* (i.e. to a vertex with no out-neighbors). By definition, vertices of *maximum* rank have their set of out-neighbors different from the set of out-neighbors of any vertex of smaller rank. This was a key insight in [10] in decomposing extensional DAGs by vertices of maximum rank.

Since essential DAGs are defined in terms of in-neighbors, we need the opposite notion of rank. Define the *depth* of a vertex x in a DAG as the length of any longest path from a *source* (i.e. from a vertex with no in-neighbors) to x. Note that a vertex of maximum depth in G must be a sink of G, but the converse does not necessarily hold. Let us denote by $d(n,k)$ the number of labeled essential DAGs with n vertices, and in which there are exactly k vertices of maximum depth. The following theorem, via its proof, explains the new decomposition of essential DAGs by vertices of maximum depth.

THEOREM 2.1. *For any $n \geq 1$, $d(n,n) = 1$ holds. Moreover, for any $k \in \{1,\ldots,n-1\}$, the following recurrence relation holds:*

$$d(n,k) = \binom{n}{k} \sum_{s=1}^{n-k} d(n-k,s)\Big(s(2^{n-k-s}-1) + (2^s-(1+s))2^{n-k-s}\Big)^k.$$

Proof. There are $\binom{n}{k}$ ways to choose the k vertices of maximum depth, and by removing them we obtain an essential DAG with $n-k$ vertices and s vertices of maximum depth, for some $s \in \{1,\ldots,n-k\}$. Each vertex x of maximum depth must have an in-neighbor among these s vertices, by definition of depth and maximality. We distinguish two cases.

First, x has precisely one in-neighbor y among these s vertices. In this case, any subset of the remaining $n-k-s$ vertices, except for the in-neighborhood of y, can act as in-neighborhood of x, when restricted to these $n-k-s$ vertices. Thus, there are $s(2^{n-k-s}-1)$ ways of choosing the in-neighborhood of x in this manner, where the -1 term accounts for the fact that x does not have zero in-neighbors among the s vertices.

Second, x has at least two neighbors among the s vertices. In this case any, subset of the remaining $n-k-s$ vertices can act as in-neighborhood of x, restricted to these $n-k-s$ vertices. This holds because no vertex among the $n-k$ vertices can have an in-neighbor among the s vertices of maximum depth, by definition of depth, maximality and acyclicity. Thus, there are $(2^s-(1+s))2^{n-k-s}$ ways of choosing the in-neighborhood of x, where the term $-(1+s)$ accounts for the fact that x does not have nor zero, nor exactly one in-neighbor among the s vertices. □

Trivially, the number of labeled essential DAGs with n vertices is obtained as $\sum_{k=1}^{n} d(n,k)$. The values of this number for small n were given in [14]. In Table 1 we present the values of $d(n,1)$ and $d(n,2)$ for small n.

We should note that even if our decomposition uses the notions of source, sink and rank, and parameterizes the counts based on them, these notions do not have immediate applications in Bayesian networks: they are technical tools useful in obtaining the final random generation procedure.

3. Random generation

Thanks to the proof of Theorem 2.1, in order to generate u.a.r. an essential DAG having $V = \{1,\ldots,n\}$ as vertex set, we can proceed recursively as in the

n	$d(n,1)$	$d(n,2)$
1	1	0
2	0	1
3	3	0
4	52	6
5	2175	430
6	254166	49035
7	72025471	14792841
8	48209379128	9973823300
9	73535410411119	15382053998676
10	251181726578388370	52716754248025365
11	1894237921184995595847	398475905881839278515
12	31226024774696489057711172	6575199987017708418484662

Table 1: Values $d(n,1)$ and $d(n,2)$, for small n.

case of extensional DAGs [7, Chapter 7]. This recursive procedure is explained formally in Algorithm 1, and works as follows.

Before calling Algorithm 1, we choose the number k of vertices of maximum depth, proportional to $\frac{d(n,k)}{\sum_{t=1}^{n} d(n,t)}$. Then, we choose u.a.r. the k vertices of maximum depth $\{v_1, \ldots, v_k\} = Z$, and call the recursive algorithm for $V \setminus Z$, with a number s of sources chosen as when choosing k. Suppose this recursive call returns an essential DAG G. It remains to add edges from the vertices of G to each v_i, such that: (i) each v_i has maximum depth in the resulting DAG; and (ii) the resulting DAG is essential.

For each $i \in \{1, \ldots, k\}$, we must choose whether v_i has exactly one in-neighbor among the vertices of maximum depth in G, or at least two in-neighbors. As such, we choose $b \in \{0, 1\}$ at random such that:

$$b = \begin{cases} 0, & \text{with probability } \dfrac{s(2^{n-k-s}-1)}{(s(2^{n-k-s}-1)+(2^s-s-1)2^{n-k-s})}; \\ \\ 1, & \text{with complementary probability.} \end{cases}$$

If $b = 0$, then we choose x u.a.r. among the vertices of maximum depth of G, and we choose u.a.r. a subset W, different from $N^-(x)$, of the other vertices of G, and set $N^-(v_i) = \{x\} \cup W$. Otherwise, if $b = 1$, we choose u.a.r. a subset W_1 of at least two elements of the vertices of maximum depth of G, and we choose u.a.r. a subset W_2 of the other vertices of G, and set $N^-(v_i) = W_1 \cup W_2$.

The correctness of this procedure follows from Theorem 2.1 and the arguments given in its proof. We obtain the following:

Algorithm 1: Generating u.a.r. an essential DAG with n vertices labeled by the elements of V, $|V| = n$, out of which k are vertices of maximum depth.

```
randomGenerationEssDAG($n, k, V$)
    if $k = n$ then return $(V, \emptyset)$;
    choose u.a.r. a $k$-subset $Z \subseteq V$;
    choose $s \in \{1, \dots, n-k\}$ with probability $\frac{d(n-k,s)}{\sum_{t=1}^{n-k} d(n-k,t)}$;
    $G :=$ randomGenerationEssDAG$(n-k, s, V \setminus Z)$;
    $X :=$ the set vertices of maximum depth of $G$;
    $Y := V(G) \setminus X$;
    $V(G) := V(G) \cup Z$;
    foreach $z \in Z$ do
        choose $b \in \{0, 1\}$ at random such that
            $b = \begin{cases} 0, & \text{with probability } \frac{s(2^{n-k-s}-1)}{(s(2^{n-k-s}-1)+(2^s-s-1)2^{n-k-s})}; \\ 1, & \text{with complementary probability;} \end{cases}$
        if $b = 0$ then
            choose u.a.r. $x \in X$;
            choose u.a.r. a subset $W$ of $Y$, different from $N^-(x)$;
            in $G$, set $N^-(z) := W \cup \{x\}$;
        else
            choose u.a.r. a subset $W_1$ of $X$ with at least 2 elements;
            choose u.a.r. a subset $W_2$ of $Y$;
            in $G$, set $N^-(z) := W_1 \cup W_2$;
    return $G$.
```

Theorem 3.1. *Algorithm 1 correctly generates u.a.r. a labeled essential DAG with n vertices, out of which k are vertices of maximum depth.*

Corollary 3.2. *A labeled essential DAG with n vertices can be generated u.a.r. by choosing $k \in \{1, \dots, n\}$ with probability proportional to $\frac{d(n,k)}{\sum_{t=1}^{n} d(n,t)}$, and calling* randomGenerationEssDAG*$(n, k, \{1, \dots, n\})$.*

4. Conclusions

Notice that even though we counted essential DAGs in a similar manner to extensional DAGs, we proceeded “bottom-up” by removing sinks (since essentiality involves in-neighborhoods), not “top-down” by removing sources, as for extensional DAGs (since extensionality involves out-neighborhoods). Since these two approaches appear hard to combine, we consider as an interesting combinatorial problem the one of counting the number of labeled DAGs that are both extensional and essential. However, if one considers *reverse essential* DAGs, as those such that for every edge (u, v) it holds that $N^+(u) \setminus \{v\} \neq N^+(v)$, then we believe the top-down approach can apply to count extensional and reverse essential DAGs, and we leave this for future work.

Acknowledgements

This work was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 851093, SAFEBIO) and by the Academy of Finland (grants No. 322595, 328877).

References

[1] S. A. Andersson, D. Madigan, and M. D. Perlman, *A characterization of Markov equivalence classes for acyclic digraphs*, Ann. Statist. **25** (1997), no. 2, 505–541.

[2] E. A. Bender, L. B. Richmond, R. W. Robinson, and N. C. Wormald, *The asymptotic number of acyclic digraphs, I*, Combinatorica **6** (1986), 15–22.

[3] E. A. Bender and R. W. Robinson, *The asymptotic number of acyclic digraphs, II*, J. Combin. Theory **44** (1988), 363–369.

[4] S. B. Gillispie and M. D. Perlman, *Enumerating Markov equivalence classes of acyclic digraph models*, Proceedings of the 17th Conference in Uncertainty In Artificial Intelligence, Morgan Kaufmann, 2001, pp. 171–177.

[5] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, New York, 1973.

[6] J. Kuipers and G. Moffa, *Uniform generation of large random acyclic digraphs*, Stat. Comput. **25** (2015), no. 2, 227–242.

[7] E. G. Omodeo, A. Policriti, and A. I. Tomescu, *On sets and graphs: Perspectives on logic and combinatorics*, Springer, 2017.

[8] R. Peddicord, *The number of full sets with n elements*, Proc. Amer. Math. Soc **13** (1962), 825–828.

[9] Jose M. Peña, *Approximate counting of graphical models via MCMC*, Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (Marina Meila and Xiaotong Shen, eds.), Proceedings of Machine Learning Research, vol. 2, PMLR, 2007, pp. 355–362.

[10] A. Policriti and A. I. Tomescu, *Counting extensional acyclic digraphs*, Inform. Process. Lett. **111** (2011), no. 3, 787–791.

[11] R. Rizzi and A. I. Tomescu, *Floating-point arithmetic for approximate counting and random generation problems*, (2013), CoRR abs/1307.2347.

[12] R. W. Robinson, *Enumeration of acyclic digraphs*, Proc. Second Chapel Hill Conf. on Combinatorial Mathematics and its Applications (Chapel Hill, N.C.), Univ. North Carolina, 1970.

[13] R. W. Robinson, *Counting labeled acyclic digraphs*, New directions in the Theory of Graphs (F. Harary, ed.), Academic Press, NY, 1973, pp. 239–273.

[14] B. Steinsky, *Enumeration of labelled chain graphs and labelled essential directed acyclic graphs*, Discrete Math. **270** (2003), no. 1-3, 266–277.

[15] B. Steinsky, *Asymptotic behaviour of the number of labelled essential acyclic digraphs and labelled chain graphs*, Graphs Combin. **20** (2004), no. 3, 399–411.

[16] S. Wagner, *Asymptotic enumeration of extensional acyclic digraphs*, Algorithmica **66** (2013), no. 4, 829–847.

Authors' addresses:

Romeo Rizzi
Department of Computer Science
University of Verona
Strada le Grazie 15, 37134, Verona, Italia
E-mail: romeo.rizzi@univr.it

Alexandru I. Tomescu
Department of Computer Science
University of Helsinki
P.O. Box 68, FI-00014, Finland
E-mail: alexandru.tomescu@helsinki.fi

Received March 24, 2021
Revised September 6, 2021
Accepted September 6, 2021