

On Markovian random networks

Yves Le Jan*

Abstract

We investigate the relation to random configurations and combinatorial maps of the Eulerian networks defined by Poissonian ensembles of Markov loops.

Keywords: Markov loops; networks; maps.

AMS MSC 2010: 60C05; 60J27; 60G60.

Submitted to ECP on August 27, 2018, final version accepted on May 16, 2019.

1 Introduction

The purpose of this note is to show that random networks, which can be defined as images of Poissonian Markov loop ensembles (also known as “loop soups”), are naturally associated to random configurations and combinatorial maps. The relation with configurations, defined as families of entering and exiting half edges attached to each vertex and coupled to form edges, follows easily from the distribution of the edge occupation field, given in [4]. The relation with random combinatorial maps is more novel and should be investigated further. In the last section, we collect a few properties of these random maps which can be deduced from [3].

We first present briefly the framework of our study, described in [3], and recall a few useful results. Consider a graph \mathcal{G} , i.e. a set of vertices X together with a set of non oriented edges E . We assume it is connected, and that there is no loop-edges nor multiple edges. The set of oriented edges, denoted E^o , is viewed as a subset of X^2 . An oriented edge (x, y) is defined by the choice of an ordering in an edge $\{x, y\}$.

Given a graph $\mathcal{G} = (X, E)$, a set of non negative conductances $C_{x,y} = C_{y,x}$ indexed by the set of edges E ($C_{x,y}$ vanishes if $\{x, y\}$ is not an edge), and a non negative killing measure κ on the set of vertices X , we can associate to them an energy (or Dirichlet form) \mathcal{E} , we will assume to be positive definite, which is a transience assumption. For any function f on X , we have:

$$\mathcal{E}(f, f) = \frac{1}{2} \sum_{x,y} C_{x,y} (f(x) - f(y))^2 + \sum_x \kappa_x f(x)^2.$$

There is a duality measure λ defined by $\lambda_x = \sum_y C_{x,y} + \kappa_x$. Let $G^{x,y}$ denote the symmetric Green’s function associated with \mathcal{E} . Its inverse equals $M_\lambda - C$ with M_λ denoting the diagonal matrix defined by λ .

*NYU-ECNU Institute of Mathematical Sciences at NYU Shanghai and Département de Mathématique, Université Paris-Sud. Orsay. E-mail: yves.lejan@math.u-psud.fr, yl57@nyu.edu

The associated λ -symmetric Markov chain is defined by the transition matrix $P_y^x = \frac{C_{x,y}}{\lambda_y}$.

We denote by μ the discrete loop measure associated with this Markov chain. μ is first defined on based loops by

$$\mu([x_0, x_1, \dots, x_{n-1}, x_0]) = \frac{1}{n} P_{x_1}^{x_0} P_{x_2}^{x_1} \dots P_{x_0}^{x_{n-1}}.$$

A loop is an equivalence class of based loops under time shift.

Given any oriented edge (x, y) of the graph, we denote by $N_{x,y}(l)$ the total number of jumps made from x to y by the loop l and set $N_x(l) = \sum_y N_{x,y}(l)$.

We have $\mu(\{l, N_x(l) = n\}) = \frac{(1-p_x)^n}{n}$ and therefore $\mu(\{l, N_x > 0\}) = -\ln(p_x)$, with $p_x = \frac{1}{\lambda_x G^{x,x}}$. We refer to chapter 2 of [3] for the proof. Note however that in [3] we considered mostly loops in continuous time, including one-point loops. From proposition 7 and formula 2-11 in [3], we get that $\mu(1 - e^{-t \hat{l}_x}) = \log(1 + t G^{x,x}) - \log(1 + \frac{t}{\lambda_x}) = \log\left(\frac{\lambda_x + t \lambda_x G^{x,x}}{\lambda_x + t}\right)$, \hat{l}_x being the sum of $N_x(l)$ independent exponential variables of mean λ_x . We check that this equals $\sum_1^\infty \frac{p_x^n}{n} \left(\frac{\lambda_x}{\lambda_x + t}\right)^n$.

The Poissonian loop ensemble \mathcal{L} is defined to be the Poisson process of loops of intensity μ . We refer to [3] for more details. Notations here are slightly different. Given any oriented edge (x, y) of the graph, we denote by $N_{x,y}(\mathcal{L})$ or simply $N_{x,y}$ the total number of jumps made from x to y by the loops of \mathcal{L} and set again $N_x = \sum_y N_{x,y}$.

We define a network to be a \mathbb{N} -valued function defined on oriented edges of the graph. It is given by a matrix k with \mathbb{N} -valued coefficients which vanishes on the diagonal and on entries (x, y) such that $\{x, y\}$ is not an edge of the graph. We say that k is Eulerian if

$$\sum_y k_{x,y} = \sum_y k_{y,x}.$$

For any Eulerian network k , we define k_x to be $\sum_y k_{x,y} = \sum_y k_{y,x}$. It is obvious that the field N defines a random network which verifies the Eulerian property.

For a finite graph, the distribution of the random network defined by \mathcal{L} was given in [4]. See also [10] for related Markov properties.

Theorem 1.1. *For any Eulerian network k ,*

$$P(N = k) = \det(I - P) \frac{\prod_x k_x!}{\prod_{x,y} k_{x,y}!} \prod_{x,y} P_{x,y}^{k_{x,y}}.$$

2 Networks and configurations

The distribution of N on Eulerian networks can be obtained differently when \mathcal{G} is finite. We introduce a specific type of configuration model.

Let \mathcal{C} be the space of configurations c defined as follows: attach to each vertex x c_x entering half edges and c_x exiting half edges, numbered from 1 to c_x . Then provide a coupling between entering and exiting half-edges in such a way that they define an oriented edge of \mathcal{G} .

For each oriented edge (x, y) , there are $c_{x,y}$ half edges exiting from x coupled with $c_{x,y}$ half edges entering in y . For each vertex x , $\sum_y c_{x,y} = \sum_y c_{y,x}$ is equal to c_x and we see that such a configuration c defines a Eulerian network \tilde{c} . Moreover each Eulerian network k is the image by this projection map $c \rightarrow \tilde{c}$ of $\frac{\prod_x (k_x!)^2}{\prod_{(x,y) \in E^o} k_{x,y}!}$ different configurations. Indeed, there are $\prod_x \frac{k_x!}{\prod_y k_{x,y}!}$ way of partitioning exiting half edges and $\prod_x \frac{k_x!}{\prod_y k_{y,x}!}$ ways of partitioning entering half edges according to the oriented edge of \mathcal{G} they define, and then $\prod_{(x,y) \in E^o} k_{x,y}!$ ways to couple them.

We can now conclude easily that:

Theorem 2.1. *The probability Q defined on \mathfrak{C} by*

$$Q(c) = \det(I - P) \frac{1}{\prod_x c_x!} \prod_{x,y} P_{x,y}^{c_{x,y}}$$

is such that, for any Eulerian network k , $Q(\tilde{c} = k) = P(N = k)$.

We say two configurations are opposite if they are exchanged by reversing the orientation of all half edges. We say they are equivalent if they can be exchanged by a circular permutation at each vertex, acting simultaneously on entering and exiting half-edges.

We can deduce from this result an expression for counting configurations with given vertex degrees:

Corollary 2.2. *Denote $\mathfrak{C}_{(i_x, x \in X)}$ the set of configurations with vertex degrees i_x : $\mathfrak{C}_{(i_x, x \in X)} = \{c \in \mathfrak{C}, \forall x \in X, c_x = i_x\}$ Then the joint exponential generating function of the cardinalities of these sets, $\sum_{(i_x, x \in X)} |\mathfrak{C}_{(i_x, x \in X)}| \prod_x \frac{s_x^{i_x}}{i_x!}$, defined for $0 \leq s_x < \frac{1}{d_x}$, is given by the inverse of the determinant of the matrix $\delta_{x,y} - s_x A_{x,y}$, A being the adjacency matrix of \mathcal{G} .*

Proof. Taking unit unit conductances and variable killing rates $1 - d_x s_x$, we rewrite $\sum_c Q(c) = 1$ in terms of cardinalities. □

Remark. A similar construction can be made with the even networks defined by the edge occupation field $N^{(\frac{1}{2})}$ of a loop ensemble $\mathcal{L}_{(\frac{1}{2})}$ of intensity $\frac{1}{2}\mu$ (cf [5]). $N_{\{x,y\}}^{(\frac{1}{2})}$ is the total number of jumps from x to y or from y to x made by the loops of $\mathcal{L}_{(\frac{1}{2})}$. The space of configurations is now the set \mathfrak{C}^{ev} of numbered \mathcal{G} -maps with an even number of incident edges at each vertex. At each vertex x there are $\frac{2k_x!}{\prod_y k_{\{x,y\}}!}$ way of partitioning the $2k_x$ incident edges and there are $\prod_{\{x,y\} \in E} k_{\{x,y\}}!$ ways to couple the numbers assigned to the edges between x and y . Each even network k is the image of $\frac{\prod_x (2k_x!)}{\prod_{\{x,y\} \in E} k_{\{x,y\}}!}$ different configurations and the distribution of these configurations is uniform conditionally to the network.

3 Configurations and Wilson algorithm

Let us recall the relation given in section 8-2 of [3] between loop ensembles and Wilson algorithm (cf [11]), for a finite graph. The choice of an arbitrary order on the vertices x_1, x_2, \dots allows to construct from a finite sequence of random walks a random spanning tree together with an independent set of loops l_i , based at x_i and included in the complement of $\{x_1, \dots, x_{i-1}\}$. Note that l_i can be empty as we are here in discrete time. The random network defined by this set of loops has the same distribution as N . Moreover, all such sets of loops defining the same network have equal probability.

In order to construct not only the network N but the Poissonian ensemble \mathcal{L} , these based loops l_i have to be randomly divided at their base points x_i . This partition is defined by a Poisson-Dirichlet distribution if we consider loops indexed by continuous time, with exponential holding time: see remark 21 in section 8-2 of [3]. As mentioned in [6] there is a discrete version of this splitting method: If l_i visits x_i n_{x_i} times, we can view each based loop l_i as a set of n_{x_i} excursions out of x_i and partition it randomly according to the exchangeable partition probability function (EPPF) $\frac{\prod_1^k (n_j - 1)!}{n_{x_i}!}$ occurring in the so-called "chinese restaurant process" ([8], sections 2-1 and 3-1). The EPPF $\frac{\prod_1^k (n_j - 1)!}{n_{x_i}!}$ is

the probability of a partition into a set of size n_1 , a set of size n_2 etc. The distribution of the block sizes, presented in exchangeable random order is $\frac{1}{k! \prod_1^k n_j}$ (cf section 2-1 in [8]).

The proof is based on the fact that for any partition $\{m_1, m_2, \dots, m_k\}$ of m , the probability that the k -tuple m_1, \dots, m_k is equal to the set $\{N_x(l), l \in \mathcal{L}\}$ in random order equals $P(|\{l \in \mathcal{L}, N_x(l) > 0\}| = k) \prod_{j=1}^k \frac{\mu(N_x(l)=m_j)}{\mu(N_x(l)>0)} = \frac{p_x}{k!} \prod_{j=1}^k \frac{(1-p_x)^{m_j}}{m_j}$. As $P(N_x(\mathcal{L}) = m) = p_x(1 - p_x)^m$, conditionally to $N_x(\mathcal{L}) = m$, this probability equals $\frac{1}{k! \prod_1^k m_j}$.

By exchangeability, performing the partition in $\{m_1, m_2, \dots, m_k\}$ based loops according to the original order in the excursion set yields the same distribution.

We then concatenate the excursions in each part according to their original order and define \mathcal{L} to be the associated set of (unbased) discrete time loops. Its distribution is uniform conditionally on the network.

Note that his construction can be related to random configurations. Let us call exit configuration a class of configurations with the same partition of exiting half edges. More precisely, an exit configuration is determined by assigning to each exiting half edge the vertex to which the oriented edge will lead, without specifying the entering half edge of that vertex. All configurations c in such a class c^{ex} have equal probability and the image of Q on the set of exit configurations \mathcal{C}^{ex} is $Q(c^{ex}) = \det(I - P) \prod_{x,y} P_{x,y}^{c^{ex},y}$. All exit configurations inducing the same network have equal probability.

Such an exit configuration c^{ex} , with the choice of an order on X , allows to construct a sample of the family of based loops l_i considered above. The first loop is based at the first vertex x_1 and visits it $c_{x_1}^{ex}$ times. It starts with the first exiting half-edge, continues to the vertex to which it is associated, then with the first half edge exiting this vertex and so on, using and taking at each vertex exiting half edges in increasing order until all have been used at x_1 . Then iterate this procedure with the remaining configuration, starting at the first vertex it contains, according to the order initially defined on X .

This correspondence is a bijection, and if we start with a random exit configuration, the distribution obtained on the sequence of based loops is the same as in Wilson's algorithm. Note for the following that we get the same distribution if exiting half edges are used in random order. Partitioning these based loops as above yields a set of unbased loops \mathcal{L}_{ex} distributed as \mathcal{L} .

An entrance configuration c^{in} is defined in an analogous way, in terms of entering half edges, assigning to each of them the vertex from which the oriented edge starts.

Similarly, we can define \mathcal{L}_{in} and we get in that way two sets of loops which are clearly independent conditionally on the network \tilde{c} .

4 Networks and combinatorial maps

The relations between configurations and Poissonian loop ensembles appears to be deeper, as shown in the following.

– Recall that a combinatorial map can be defined as a graph, with possibly multiple edges between vertices, equipped with a combinatorial embedding, i.e. a cyclic ordering of edges around each vertex (see [7], [1]). This ordering allows to define combinatorial faces (termed as facial walks in section 4-1 of [7]) as cycles of the permutation ϕ on oriented edges obtained by composing the edge orientation reversal ρ by the shift defined by the cyclic order. Note that the facial walks may visit a vertex several times. The map can be drawn on a surface with possibly several connected components (see [7], p. 85). Let us say it is a \mathcal{G} -map if its vertex set is contained in X and if its edges are multiple copies of elements of E (if \mathcal{G} is complete, the second condition is redundant). Let us say a map is numbered if a first edge is chosen at every vertex.

– An element c of \mathcal{C} defines a numbered \mathcal{G} -map $\mathcal{M}(c)$ with an even number of incident edges at each vertex. The first edge at each vertex is determined by the first exiting half edge and the half edge it is coupled to, the second by the first entering half edge and the half-edge it is coupled to, the third by the second exiting half edge and the half-edge it is coupled to, and so on, alternating entering and exiting half edges according to their original cyclic order. We actually get in this way, an orientation on the edges of the map. At each vertex, the cyclic order on edges alternates exiting and entering edges. The number of oriented edges carried by (x, y) is equal to $\tilde{c}_{x,y}$. Two equivalent configurations define the same \mathcal{G} -map.

– An equivalent class of configurations defines also alternating signs on the combinatorial faces of its associated map.

The positive faces \mathcal{F}_+ are defined by the cycles of the permutation on edges obtained by mapping each exiting half-edge with the exiting half-edge following, (in the above defined cyclic order at each vertex), the entering half-edge coupled to it. Oriented face contours are defined by the orientation of their edges.

The negative faces \mathcal{F}_- are defined by the cycles of the permutation on edges obtained by mapping an entering half-edge with the entering half-edge following the exiting half-edge coupled to it. Oriented face contours are defined by reversing the orientation of their edges.

Taking the opposite configuration exchanges \mathcal{F}_+ and \mathcal{F}_- .

Each edge of the map is in this way adjacent to two faces of opposite signs, and giving it opposite orientations as part of a face contour. Note that \mathcal{F}_+ determines the configuration, up to equivalence. The same holds for \mathcal{F}_- .

The image on \mathcal{G} of the sets of oriented face contours of \mathcal{F}_+ and \mathcal{F}_- are multisets of loops denoted $(\mathcal{L}_+$ and \mathcal{L}_-).

– We can now state our main result:

Theorem 4.1. *The loop ensembles \mathcal{L}_+ and \mathcal{L}_- defined by a random configuration under the probability Q have both the same distribution as \mathcal{L} .*

Proof. Note first that (in contrast with the original version of the algorithm presented in the previous section), this construction does not require additional independent partitions.

We can again choose an arbitrary order on the vertices to run a different version of Wilson algorithm to construct \mathcal{L}_+ . This algorithm can be viewed as a way, starting from c^{ex} (i.e. the partition of exiting half edges) to sample the random coupling with entering edges progressively. This determines a configuration c sampled uniformly in c^{ex} . We start a based loop at the first vertex with the first exiting half edge, then move to the vertex attached to it by the partition, and choose an entering half edge to couple it. Then follow the exiting half edge following this entering edge in the alternating cyclic order, and so on until we reach again the base point. Then, a based loop is created if the first exiting half-edge follows the last entering one, in the cyclic order. We iterate the process until this happens, and create a based loop by concatenating the excursions we constructed. Note that conditioning on the initially given partition of exiting half edges and on the partial coupling with entering half edges done until a return time (this includes the excursions out of x_1 up to this return time), the creation of a based loop occurs with probability one over the number of unused entering half edges at the base point. Then restart at the same vertex with the next free exiting half-edge. Iterate until all half edges at the first vertex have been used.

– We can then restart the algorithm at the next vertex with unused exiting half-edge, starting with the first one. Iterate until all half edges of all vertices have been used. If we start with a random initial exit configuration, the concatenation of the excursions

obtained at each vertex has the same distribution as the family of based loops $\{l_i\}$ defined by Wilson algorithm as explained at the end of the previous section (the exiting half edges here are used in random order). Conditionally on these based loops, the random ordered partitions of them we have obtained at each vertex are exactly the size-biased presentation of the random partition defined by the “chinese restaurant process” (see section 2-1 in [8]). This observation completes the proof of the theorem as the case of \mathcal{L}_- can be treated in the same way. \square

5 Expectations calculations

The study of the geometrical properties of the random combinatorial maps defined by random configurations should be pushed further. In this last section, we present a few results in this direction.

Using the results of [3], expressions for the expected number of vertices, edges and combinatorial faces of $\mathcal{M}(c)$ can be given. These are respectively: $E(|\{x \in X, c_x > 0\}|) = \sum_x (1 - \frac{1}{\lambda_x G^{x,x}})$ (as N_x follows a geometric distribution of parameter $\frac{1}{\lambda_x G^{x,x}}$ (see the remark following formula 4-3 in [3]), $E(\sum_{x,y} \tilde{c}_{x,y}) = \sum_{x,y} C_{x,y} G^{x,y}$ (see formula 2-10 in [3]) and $E(|\mathcal{L}_+(c)| + |\mathcal{L}_-(c)|) = -2 \ln \det(I - P)$ (see formula 2-5 in [3]).

Similar expressions can be given for the variances of the vertices and edges numbers. Much more intricate expressions can also be given for the number and the sizes of the connected components of the map (see [2]).

All these results apply in particular to the case of complete graphs, which is of special interest in relation with the theory of combinatorial maps.

For d vertices and constant killing rate κ , $\lambda = d - 1 + \kappa$ and $G = \frac{1}{d+\kappa}(I + \frac{1}{\kappa}J)$, J denoting the (d, d) matrix with all entries equal to 1. Setting $p = \frac{1}{d-1+\kappa}$, $\det(I - P) = (1 - (d - 1)p)(1 + p)^{d-1}$.

Inserting this in the expressions recalled hereinabove, it appears that if we let d and κ increase to infinity, with $\frac{\kappa}{d}$ converging to zero, the expected number of vertices and the expected number of edges are both equivalent to $\frac{d}{\kappa}$. More precisely, they are respectively equal to $\frac{d}{(\kappa+1)} + 1 + o(1)$ and $\frac{d}{\kappa} - 1 + o(1)$. The expected number of faces is $2 \ln(d/\kappa) - 2 + o(1)$.

Following section 4-1 in [7], the Euler characteristic $\chi(c)$ of the combinatorial map $\mathcal{M}(c)$ is defined as the number of vertices + the number of combinatorial faces - the number of edges, i.e $|\{x \in X, c_x > 0\}| + |\mathcal{L}_+(c)| + |\mathcal{L}_-(c)| - N(c)$ with $N(c) = \sum_{x,y} \tilde{c}_{x,y}$.

Its expectation can be expressed as

$$E(\chi) = \sum_x (1 - \frac{1}{\lambda_x G^{x,x}}) - 2 \ln \det(I - P) - \sum_{x,y} C_{x,y} G^{x,y}.$$

In particular, for the complete graph, $E(\chi) = d(1 - (1 - \frac{1}{\kappa+1})(1 + \frac{1}{d-1+\kappa})) - \frac{d(d-1)}{\kappa(d+\kappa)} - 2 \ln(\frac{\kappa}{d-1+\kappa}) - 2(d - 1) \ln(1 + \frac{1}{d-1+\kappa})$. The expectation of the characteristic χ can converge to any value v by taking $\kappa(d, v) = \sqrt{\frac{d}{u(d, v)}}$ with $u(d, v) > 1$ defined as the unique solution of the equation $u(d, v) - \ln(u(d, v)) = \ln(d) - v$. Note that $u(d, v)$ is equivalent to $\ln(d)$ as d increases to infinity. If we take $\kappa(d) = \sqrt{\frac{d}{\ln(d)}}$, the expectation of χ is equivalent to $\ln(\ln(d))$ as d increases to infinity.

Note however that the number of faces in the 2-cell embedding in surfaces defined by the combinatorial map is larger than the number of facial walks. Hence the sum of the Euler characteristics of the surfaces is smaller or equal to $\chi(c)$ (see proof of 3-4-1 in [7]). The average number of vertices visited by a facial walk is $\frac{\sum_x \mu(N_x > 0)}{|\mu|} =$

$\frac{-\sum_x \ln(P(N_x=0))}{-\ln \det(I-P)} = \frac{\sum_x \ln(\lambda_x G^{x,x})}{-\ln \det(I-P)}$ It is smaller from the average length of the walk which is $\frac{\sum_x \mu(N_x)}{|\mu|} = \frac{\sum_x (\lambda_x G^{x,x} - 1)}{-\ln \det(I-P)}$. The difference is the average number of multiple points on the loop, counted with their multiplicities minus 1.

For the complete graph, with d and κ increasing to infinity, with $\frac{\kappa}{d}$ converging to zero, we get that the average number of multiple points is equivalent to $\frac{d}{2\kappa^2 \ln(\frac{d}{\kappa})}$ which converges to $\frac{1}{2}$ if we take $\kappa(d) = \sqrt{\frac{d}{\ln(d)}}$.

Consider finally essential vertices, i.e. such that $N_x > 1$. Their expected number in general is $\sum_x (1 - \frac{1}{\lambda_x G^{x,x}})^2$ which equals $\frac{d(d+2\kappa-1)^2}{(\kappa+1)^2(d-1+\kappa)^2}$ in the complete graph case. If we take $\kappa(d) = \sqrt{\frac{d}{\ln(d)}}$ or $\kappa(d, v) = \sqrt{\frac{d}{u(d, v)}}$, the expected number of facial walks and the expected number of essential vertices are both equivalent to $\ln(d)$ as d increases to infinity.

References

- [1] Sergei K. Lando and Alexander K. Zvonkin. *Graphs on Surfaces and their Applications*. Springer, Berlin-Heidelberg (2004). MR-2036721
- [2] Yves Le Jan and Sophie Lemaire. Markovian loop clusters on graphs. arXiv:1211.0300. *Illinois Journal of Mathematics* Volume 57, 525–558 (2013). MR-3263044
- [3] Yves Le Jan. Markov paths, loops and fields. *École d’Été de Probabilités de Saint-Flour XXXVIII* (2008). *Lecture Notes in Mathematics* 2026. Springer, Berlin-Heidelberg (2011). MR-2815763
- [4] Yves Le Jan. Markov loops, free field and Eulerian networks. arXiv:1405.2879. *J. Math. Soc. Japan* 67, 1671–1680 (2015). MR-3417508
- [5] Yves Le Jan. Markov loops, coverings and fields. *Ann. Fac. Sci. Toulouse Math.* 26, 401–416 (2017). MR-3640896
- [6] Yves Le Jan. Markov loops topology. arXiv:1707.05106.
- [7] Bojan Mohar, Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins Studies in the Mathematical Sciences (2001). MR-1844449
- [8] Jim Pitman. *Combinatorial Stochastic Processes*. 32th St Flour Summer School. *Lecture Notes in Math.* 1875. Springer, Berlin-Heidelberg (2006). MR-2245368
- [9] Remco Van der Hofstad. *Random Graphs and Complex Networks*. Cambridge Series in Statistical and Probabilistic Mathematics (2017). MR-3617364
- [10] Wendelin Werner. On the spatial Markov property of soups of unoriented and oriented loops. *Séminaire de de Probabilités XLVIII*, 481–503. *Lectures Notes in Math.* 2168. Springer Berlin-Heidelberg (2016). MR-3618142
- [11] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. *Proceedings of the 28th annual ACM symposium on Theory of computing*, 296–303. ACM (1996). MR-1427525

Acknowledgments. Thanks are due to the referees for suggesting several improvements in this paper.

Electronic Journal of Probability

Electronic Communications in Probability

Advantages of publishing in EJP-ECP

- Very high standards
- Free for authors, free for readers
- Quick publication (no backlog)
- Secure publication (LOCKSS¹)
- Easy interface (EJMS²)

Economical model of EJP-ECP

- Non profit, sponsored by IMS³, BS⁴, ProjectEuclid⁵
- Purely electronic

Help keep the journal free and vigorous

- Donate to the IMS open access fund⁶ (click here to donate!)
- Submit your best articles to EJP-ECP
- Choose EJP-ECP over for-profit journals

¹LOCKSS: Lots of Copies Keep Stuff Safe <http://www.lockss.org/>

²EJMS: Electronic Journal Management System <http://www.vtex.lt/en/ejms.html>

³IMS: Institute of Mathematical Statistics <http://www.imstat.org/>

⁴BS: Bernoulli Society <http://www.bernoulli-society.org/>

⁵Project Euclid: <https://projecteuclid.org/>

⁶IMS Open Access Fund: <http://www.imstat.org/publications/open.htm>