Algorithms Seminar 2000–2001,
F. Chyzak (ed.), INRIA, (2002), pp. 45–48.

Available online at the URL
`http://algo.inria.fr/seminars/`.

Random Walks and Heaps of Cycles

Philippe Marchal

Département de mathématiques et applications, École normale supérieure (France)

April 23, 2001

Summary by Cyril Banderier

Abstract

The problem addressed here is the covering time of random walks on a graph satisfying "self-avoiding" properties. Appealing to the combinatorics of heaps of cycles, the author derives explicit expressions of the laws for several algorithms related to loop-erased random walks (and thus to spanning trees and Hamiltonian cycles samplings), Łukasiewicz walks, and taboo random walks.

1. Spanning Trees, Hamiltonian Cycles, Spanning Heaps of Cycles

Combinatorial tools (such as generating functions, context-free grammars) generally have too little "memory" to deal with "self-avoiding" walks, and thus their enumeration remains a widely open problem. However, for a few years, an approach via loop-erased random walks has seemed promising (see [3] and also the summary of R. Kenyon's talk in the proceedings of years 99–00). Philippe Marchal exploits here the theory of determinants related to properties of heaps of cycles[1] and then gives the average time needed to generate self-avoiding walks of several kinds.

Define a *cycle* as a path beginning and ending at the same point, and not containing any subcycle.

Given a connected graph G (where each edge is oriented and weighted), one wants to find

- a spanning tree T of this graph (i.e., a tree T whose each edge is an edge from G and each vertex of G a node of T);
- a Hamiltonian cycle C (i.e., a cycle C whose each edge is an edge from G, and each vertex of G is visited exactly one time by C);
- a spanning heap H of cycles (i.e., a heap H of cycles whose each edge of is an edge from G, and each vertex of G is visited by at least one of the cycles of H).

In order to get a spanning tree or a Hamiltonian cycle of the graph, it is interesting to use probabilistic algorithms, since these problems are NP-complete.

On the connected edge-weighted oriented graph G (the weights are given by a matrix P), one considers the Markov chain $(X_n)_{n\in\mathbb{N}}$, defined by

$$\mathbf{P}(\, X_{n+1} = i \mid X_n = j\,) = P_{ij}.$$

This means that the probability to go from vertex i (where you are at time n) to vertex j is the weight P_{ij} of edge (i,j). In this talk, one considers irreducible Markov chains only (i.e., the random walk visits each of the m vertices of the graph G an infinite number of times with probability 1) so that there always exists a vertex-stationary distribution $(\pi_1,\dots,\pi_m)$, where π_j is the probability

[1] The cycle decomposition was introduced by Cartier and Foata [2] and the modelling via heaps of cycles is due to Viennot [8].

to be at vertex j, after a long enough time. Similarly, there is an edge-stationary distribution for the edges.

Define the weight of the tree T (resp. the cycle C, the heap H) as the product of the weights of its edges. Consider now a trajectory (a realization) of the Markov chain $(X_n)_{n\in\mathbb{N}}$, and whenever one performs a cycle, one erases this cycle from the walk and one puts this cycle on a heap (by construction, this cycle has no subcycle). If one stops at time n, one gets a "loop-erased random walk" (which is a self-avoiding walk of length less than or equal to n) and a heap of cycles. It will be explained in Section 3 how to use this loop-erased random walk to get a spanning tree, a Hamiltonian cycle or a spanning heap of cycles.

2. Generating Functions

Let N_{ij} be the number of visits through the edge (i,j) and t_{ij} a formal variables associated to the edge (i,j). Note that N_{ij} takes also into account the visits in the cycles that get erased, thus $\sum N_{ij} = n$ is the length of the walk. Then, define the formal weight function $\widetilde{w}$ as the function which transforms a path (i.e., a sequence of edges) $\gamma = \big((x_0,x_1),\dots,(x_{n-1},x_n)\big)$ into the polynomial

$$\widetilde{w}(\gamma) = \prod_{i=1}^{n} P_{x_{i-1}x_i} t_{x_{i-1}x_i}.$$

This definition (as a product of the formal weights of each edge) is easily extended to trees, cycles, graphs. Define now the formal transition matrix $\widetilde{P}$ by $\widetilde{P}_{ij} = P_{ij}t_{ij}$ and, for a subset S of the edges of the graph G, define $\widetilde{P}_S$ as equal to $\widetilde{P}$ excepted that $(\widetilde{P}_S)_{ij} := 0$ whenever $i \notin S$ or $j \notin S$.

Let $\mathcal{C}$ be the set of cycles and $\mathcal{H}$ the set of heaps of cycles from $\mathcal{C}$, then

$$\sum_{H\in\mathcal{H}} \widetilde{w}(H) = \left(\sum_{k\geq 1}\sum_{C_1\ldots C_k\in\mathcal{C}} (-1)^k \widetilde{w}(C_1)\ldots\widetilde{w}(C_k)\right)^{-1} = \frac{1}{\det(\mathrm{Id}-\widetilde{P})}$$

where C_1, $\dots$, C_k are disjoint cycles belonging to $\mathcal{C}$. The proof comes from an expansion of the determinant as a sum over all permutations and then decompose each permutation in a product of cycles (each $(-1)^k$ is nothing but an avatar of the signature of each permutation).

If $\mathcal{H}$ stands for the set of heap of cycles avoiding a subset S of the edges of the graph G, one has

$$\sum_{H\in\mathcal{H}} \widetilde{w}(H) = \frac{1}{\det(\mathrm{Id}-\widetilde{P}_S)}.$$

Whereas if $\mathcal{H}$ stands for the set of heaps of cycles intersecting a set S, one has

$$\sum_{H\in\mathcal{H}} \widetilde{w}(H) = \frac{\det(\mathrm{Id}-\widetilde{P}_S)}{\det(\mathrm{Id}-\widetilde{P})}.$$

For example, if one stops the random walk X as soon as it reaches a given point v and one considers the associated loop-erased walk γ, one has the following probability generating function

$$\mathbf{E}\left(\prod_{(i,j)} t_{ij}^{N_{ij}}, \gamma\right) = \frac{\widetilde{w}(\gamma)}{\det(\mathrm{Id}-\widetilde{P}_v)}.$$

The right member has to be read as a generating function in several variables (the number of edges in G) whose coefficient, e.g., $[t_{1,2}^4\dots t_{1,4}^0\dots t_{3,5}^7] = \frac{\widetilde{w}(\gamma)}{\det(\mathrm{Id}-\widetilde{P}_v)}$, gives the probability that the random walk X visits edge $(1,2)$ 4 times, edge $(1,4)$ 0 time, edge $(3,5)$ 7 times, ... while the associated loop-erased random walk finally gives γ.

Remark: For queuing theory, assurance, etc., a usual model is left-continuous random walks (walks on $\mathbb{Z}$ with a finite set of jumps where the only negative jump is -1). These walks are sometimes called Łukasiewiecz walks, due to their correspondence with simple families of trees, their nice combinatorial and analytic properties are well understood, see [1]. Let $p_i,\ \ i \geq -1$, be the probability to do a jump i and let P_n be the transition matrix restricted to $[0, n]$. Then $D_n(t) := \det(\mathrm{Id} - tP_n)$ can easily be computed by the following recurrence:

$$D_0(t) = D_{-1}(t) = 1, \qquad D_k(t) = D_{k-1}(t) - \sum_{n=0}^{k} p_n t (p_{-1} t)^n D_{k-n-1}(t).$$

3. Wilson's Algorithm and Some Variants

By convention, one considers spanning trees whose edges are all oriented to the root. Let $\mathcal{T}_r$ be the set of spanning trees rooted at r; a well-known result, the matrix-tree theorem implies that

$$\frac{\sum_{T \in \mathcal{T}_r} w(T)}{\pi_r} = \text{constant}.$$

The striking fact is that the quotient does not depend on r.

Wilson's algorithm [7] allows to construct a random spanning tree with a given root r. Specify an arbitrary order on G. Start the loop-erased random walk from the first point (with respect to the above order) until it reaches r. It gives a self-avoiding walk T_1. Then, restart from the first remaining point until one reaches T_1, one got a subtree T_2, etc. Finally, one gets a random spanning tree, rooted at r.

The probability to get this tree T is proportional to its weight $w(T)$ and does not depend on the chosen order. The proof relies on the correspondence between trajectories and heap of cycles as explained above. The probability generating function is

$$\mathbf{E}\left(\prod_{(i,j) \in G^2} t_{ij}^{N_{ij}},\ T \right) = \frac{\widetilde{w}(T)}{\det(\mathrm{Id} - \widetilde{P}_r)}$$

and thus the average time is $\mathrm{tr}\big((\mathrm{Id} - P_r)^{-1}\big)$.

Similarly, one can get a Hamiltonian cycle. Start the loop-erased random walk from a point $r \in G$ and stop the first time one gets a Hamiltonian cycle C in the heap of cycles. Let $\mathcal{C}$ be the set of Hamiltonian cycles. Then the probability generating function is independent from r:[2]

$$\mathbf{E}\left(\prod_{(i,j) \in G^2} t_{ij}^{N_{ij}},\ C \right) = \frac{\widetilde{w}(C)}{\det(\mathrm{Id} - \widetilde{P}) + \sum_{C' \in \mathcal{C}} \widetilde{w}(C')}.$$

Finally, one gets also a sampling algorithm for a spanning heap of cycles. Choose an arbitrary order on G. Start the loop-erased random walk from a_1, stop when it returns to a_1. Then consider the first remaining non-visited point a_2 and start a loop-erased random walk from a_2 and stop when it returns to a_2, etc. Stop when all the points have been visited. Here again, the occupation measure does not depend on the chosen order. The proof relies on the fact that one gets a minimal

[2]Consider a nearest neighbor random walk on a cyclic graph with m vertices, and stop the walk when it comes back to the starting point, after having covered all the graph. Then, the occupation measure does not depend on the starting point. This phenomenon was observed by Pitman in 1996 for Brownian motion.

spanning heap. The probability generating function is

$$\mathbf{E}\left(\prod_{(i,j)\in G^2} t_{ij}^{N_{ij}}, H\right) = \det(\mathrm{Id} - \widetilde{P}) \sum_{F\subset G} \frac{(-1)^{|F|}}{\det(\mathrm{Id} - \widetilde{P}_F)}.$$

The waiting time W of the algorithm is stochastically less than the first time W_v that the walk returns to vertex v, *after* having visited all the vertices of the graph:

$$\forall v \in G, \forall n \in \mathbb{N} \qquad \mathbf{P}(W \leq n) \geq \mathbf{P}(W_v \leq n)$$

The proof follows from the fact that any spanning pyramid (see [6]) contains a minimal spanning tiling. The author also derives this inequality:

$$\frac{1}{\inf_{v\in G} \pi_v} \leq \mathbf{E}(W) \leq \sum_{v\in G} \frac{1}{\pi_v}.$$

4. Killed Random Walks

Let $q \in (0,1)$, to kill X with a probability $1-q$ means to add a sink s and to put some probabilities of transition $P'_{ij} = qP_{ij}$, $P'_{is} = 1-q$. Then, if one runs Wilson's algorithm (rooted at s), one gets a random heap with a probability proportional to $\widetilde{w}(H)q^{|H|}$ where $|H|$ is the number of edges in H. The following process also provides a random heap (with the same distribution): construct an infinite random heap and then color each edge in red with probability q. Drop the red cycles. Then one gets a red heap with the wanted probability and another heap whose all minimal cycles have at least one non-colored edge. Let q vary continuously and thus obtain an increasing family of heaps. At a given value q, an upside-down pyramid falls. The probability that an upside-down pyramid P falls between q and $q+dq$ equals

$$\widetilde{w}(P)q^{|P|-1}dq.$$

Some generalisations of this idea allow to generate walks constrained to avoid a specified set, known as taboo random walks.

This summary is related to Marchal's articles [4, 5, 6]. The readers who want to learn more about "Perfectly Random Sampling with Markov Chains" can have a look at the web site maintained by David Wilson at `http://dimacs.rutgers.edu/~dbwilson/exact/`.

Bibliography

[1] Banderier (Cyril). – *Combinatoire analytique des chemins et des cartes.* – Thèse universitaire, Université de Paris VI, 2001.

[2] Cartier (P.) and Foata (D.). – *Problèmes combinatoires de commutation et réarrangements.* – Springer-Verlag, Berlin, 1969, iv+88p.

[3] Lawler (Gregory F.). – Loop-erased random walk. In *Perplexing problems in probability*, pp. 197–217. – Birkhäuser Boston, Boston, MA, 1999.

[4] Marchal (Philippe). – Cycles hamiltoniens aléatoires et mesures d'occupation invariantes par une action de groupe. *Comptes Rendus de l'Académie des Sciences. Série I. Mathématique*, vol. 329, n° 10, 1999, pp. 883–886.

[5] Marchal (Philippe). – Loop-erased random walks, spanning trees and Hamiltonian cycles. *Electronic Communications in Probability*, vol. 5, 2000, pp. 39–50.

[6] Marchal (Philippe). – *Loop-erased random walks and heaps of cycles.* – Prépublications n° DMA-01-07, École normale supérieure, Paris, 2001.

[7] Propp (James Gary) and Wilson (David Bruce). – How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, vol. 27, n° 2, 1998, pp. 170–217. – 7th Annual ACM-SIAM Symposium on Discrete Algorithms (Atlanta, GA, 1996).

[8] Viennot (Gérard Xavier). – Heaps of pieces. I. Basic definitions and combinatorial lemmas. In *Combinatoire énumérative (Montréal, Québec, 1985)*, pp. 321–350. – Springer, Berlin, 1986.