Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 59 (2017) 37–50

www.elsevier.com/locate/endm

An optimal algorithm to generate extendable self-avoiding walks in arbitrary dimension

Pascal Préa [1,2]

École Centrale Marseille
Laboratoire d'Informatique Fondamentale de Marseille
LIF, CNRS UMR 7279
Marseille, France

Mathieu Rouault [3]

Department of Oceanography, MARE Institute
and
Nansen-Tutu Centre for Marine Environmental Research
University of Cape Town
Cape Town, Republic of South Africa

François Brucker [4]

École Centrale Marseille
Laboratoire d'Informatique Fondamentale de Marseille
LIF, CNRS UMR 7279
Marseille, France

Abstract

A self-avoiding walk (SAW) is *extendable* [10,13] if it can be extended into an infinite SAW. We give a simple proof that, for every lattice, extendable SAWs admit the same connective constant as the general SAWs and we give an optimal linear algorithm to generate random extendable SAWs. Our algorithm can generate every

http://dx.doi.org/10.1016/j.endm.2017.05.004

extendable SAW in dimension 2. For dimension $d > 2$, it generates only a subset of the extendable SAWs. We conjecture that this subset is “large” and has the same connective constant as the extendable SAWs. Our algorithm produces a kinetic distribution of the extendable SAWs, for which the critical exponent ν is such that $\nu \approx .57$ for $d = 2$, $\nu \approx .51$ for $d = 3$ and $\nu \approx .50$ for $d = 4, 5, 6$.

Keywords: self-avoiding walk, connective constant, critical exponent ν, random generation.

1 Introduction

A walk on a lattice is *self-avoiding* if it never passes twice through the same vertex (see Figure 1). Self-avoiding walks (SAWs) appeared as a model for polymers [7]. They also have applications in statistical physics [8] and in probability theory [15].

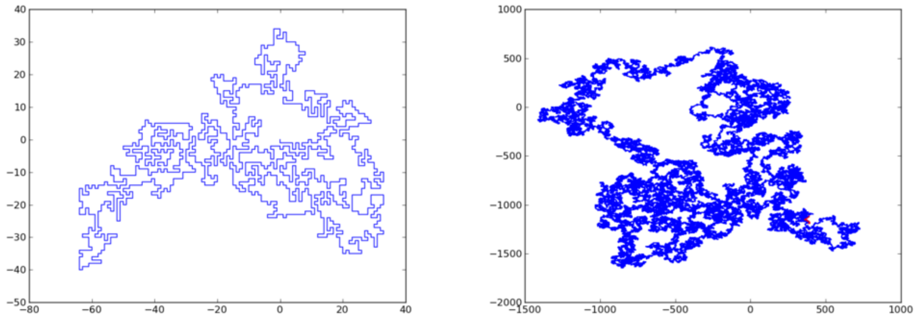

Fig. 1. A SAW of length 1751 and a SAW of length $5 \cdot 10^5$ on the square lattice in dimension 2

Formally speaking, for instance on the square grid $\mathbb{Z} \times \mathbb{Z}$, a *walk* is a sequence $W = (w_0, w_1, \dots w_n)$ of vertices such that, for all $i < n$, w_i and w_{i+1} are neighbors (i.e. if $w_i = (x_i, y_i)$ and $w_{i+1} = (x_{i+1}, y_{i+1})$, then ($x_{i+1} = x_i$ and $y_{i+1} = y_i \pm 1$) or ($y_{i+1} = y_i$ and $x_{i+1} = x_i \pm 1$)). The walk W is *self-avoiding* if $i \neq j \Rightarrow w_i \neq w_j$.

[1] Part of this work was done while one of the authors (P.P.) was visiting the University of Cape Town.

[2] Email: `pascal.prea@lif.univ-mrs.fr`
[3] Email: `mathieu.rouault@uct.ac.za`
[4] Email: `francois.brucker@lif.univ-mrs.fr`

Despite a very simple and natural definition, and although they are very close to random walks (which are standard objects, very well studied and known), self-avoiding walks suggests many problems, both theoretical and practical.

Given a lattice (for instance, the hexagonal lattice in the plane or the cubic lattice $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$), the two main theoretical questions concerning SAWs on that lattice are:

- What is the number c_n of SAWs of length n?
- Given a distribution on the SAWs (for instance the uniform distribution on the SAWs of length n), what is the average Euclidean distance d_n between the two extremities of a SAW of length n?

It is conjectured [15] that:

- $c_n \sim \mu^n n^{\gamma-1}$
- $d_n \sim n^{\nu}$

It is strongly believed that the *critical exponents* γ and ν depend only on the dimension (and not on the lattice), while the *connective constant* μ also depends on the lattice. It is known that the connective constant (which is equal to $\lim_{n\to\infty} c_n^{1/n}$) exists for every lattice. Despite decades of efforts, these questions remain largely unsolved, even in low dimensions. It has recently been shown [6] that the connective constant for the hexagonal lattice in the plane is $\sqrt{2+\sqrt{2}}$. Conversely, for the square lattice in the plane, it is conjectured [11] that the connective constant is the unique positive root of $13x^4 - 7x^2 - 581 = 0$ (i.e. $\sqrt{\frac{7+\sqrt{30261}}{26}} \approx 2.64$).

For two-dimensional lattices, it is conjectured that $\gamma = 43/32$ and that, for the uniform distribution, $\nu = 3/4$.

From a practical point of view, it is very interesting to generate random self-avoiding walks since these walks arise as models for various physical phenomena. In addition, generating random self avoiding walks yields an approximation for the critical exponents and the connective constant. There are many algorithms to generate random self-avoiding walks; among them the pivot algorithm [14,12], the Berretti-Sokal algorithm [1], the Rosenbluth algorithm [19,9,18], flat-PERM [5]. All these algorithms are Monte Carlo algorithms [17].

The (apparently) simplest way to generate a random SAW consists of choosing, at each step, a neighbor of the courant end of the walk, not already in the walk, and to add it to the walk. When the last point of the walk has no “free neighbors”, we can stop or use backtracking. Backtracking needs prohibitive computer resources, and, in dimension 2, not using it yields only short paths (see Figure 2).

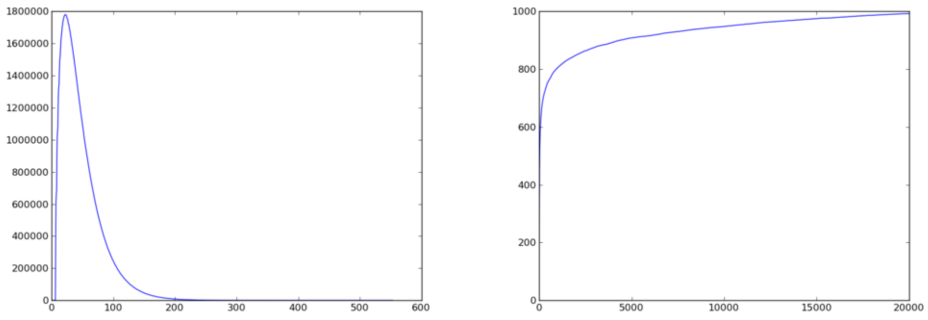

Fig. 2. Histogram of length without using backtracking (left) and by using backtracking (Right) for SAWs on the planar lattice $\mathbb{Z} \times \mathbb{Z}$. The left histogram was obtained with 10^8 trials. On the x-axis are the lengths, and on y-axis the number of SAWs having these lengths. The right histogram was obtained with 10^5 trials. On the x-axis there is the number of steps ($\approx$ the time), with scale 1/100, and on the y-axis the average length at that moment.

In this paper we study a special case of self-avoiding walks : the *extendable SAWs* [10,13]. In Section 2 we give their definition and some properties, and in Section 3 we give an optimal $O(n)$ algorithm to generate a random extendable SAW of length n. An implementation of this algorithm can be found at `https://github.com/pascalprea/Extendable_SAW`.

2 Definitions and properties

We say that a SAW $W = (w_0, w_1, \dots w_n)$ is *extendable* if W can be extended into an infinite SAW (i.e. if there exists an infinite SAW $W_\infty = (w_0, w_1, \dots w_n, \dots)$).

Proposition 2.1 *For every regular lattice of the Euclidean space (e.g. the triangular lattice in the plane or the cubic lattice in the 3-dimensional space), extendable SAWs admit a connective constant which has the same value as the connective constant for general SAWs.*

This result is a particular case of Theorem 1 in [10]; it admits a very simple proof that we give here.

Proof. We first show that extendable SAWs admit a connective constant μ_e. The proof is exactly the same as for general SAWs (see [15] p. 9-10); we will only give a sketch. Let c_n^e be the number of extendable SAWs of length n. Every extendable SAW of length $n+m$ is the concatenation of an extendable SAW of length n and of an extendable SAW of length m. So, $c_{n+m}^e \leq c_n^e \times c_m^e$, thus $\log c_{n+m}^e \leq \log c_n^e + \log c_m^e$ and $\mu_e = \lim_{n\to\infty}(c_n^e)^{1/n}$ exists.

We now prove that $\mu_e = \mu$. Let $W = (w_1, w_2, \ldots, w_n)$ be a SAW of length n, $W^\star = (w_1, w_2, \ldots w_{\lfloor \sqrt[3]{n} \rfloor})$ is an extendable SAW (otherwise, the SAW $W^+ = (w_{\lfloor \sqrt[3]{n} \rfloor}, \ldots, w_n)$ would be entirely included in a finite volume whose surface is $W^\star$; but if the dimension of the space is > 1, any volume has a surface greater than its cubic root). Thus $c_n \leq c^e_{\sqrt[3]{n}} \cdot c_{n-\sqrt[3]{n}}$; so, for n large enough, $\mu^n \leq \mu_e^{\sqrt[3]{n}} \cdot \mu^{n-\sqrt[3]{n}}$ and $\mu_e \geq \mu$. As $\mu_e \leq \mu$, the property is proved. □

Given an extendable SAW $W = (w_0, w_1, \ldots w_n)$, an *Ariadne thread* (associated with W) is an infinite SAW $A = (a_0, a_1, \ldots)$ such that:

- a_0 and w_n are neighbors,
- A and W do not cross ($\forall i, j, w_i \neq a_j$).

Equivalently, the concatenation $WA = (w_0, w_1, \ldots, w_n, a_0, a_1, \ldots)$ is an infinite SAW; so, a self-avoiding walk is extendable if and only if it admits an Ariadne thread.

3 An optimal algorithm to generate random extendable walks

3.1 Description of the algorithm

Our algorithm constructs a SAW step by step, starting with the empty walk and adding a point at each step. The main idea consists of maintaining, at each step, an Ariadne thread. We will describe it for the hypercubic lattice $\mathbb{Z}^d$.

If $n = 0$, every axis-parallel half line starting at w_0 is an Ariadne thread for W. If $n > 0$, the SAW $W_{n-1} = (w_0, w_1, \ldots w_{n-1})$ admits an Ariadne thread $A_{n-1} = (a_0, a_1, \ldots)$. The construction/modification of the Ariadne thread will depend on the configuration at the last point w_n of W (we suppose that W is an extendable SAW, and so $w_n \notin W_{n-1}$).

(i) There exists an axis-parallel half line $L = (a_0 = w_n, a_1, \ldots)$ which does not cross W_{n-1} (we say that w_n *sees the infinite*). In this case, $A = (a_1, a_2, \ldots)$ is an Ariadne thread for W.

(ii) The point w_n is on A_{n-1}, i.e. there exists i such that $w_n = a_i$. In this case, $A = (a_{i+1}, a_{i+2}, \ldots)$ is an Ariadne thread for W.

(iii) The point w_n does not see the infinite and is not on A_{n-1}. There exists a path P avoiding W_{n-1} between w_n and A_{n-1}. For algorithmic efficiency, we only search this path in a neighborhood of w_{n-1} (namely the points at L_∞-distance 1 from w_{n-1}). The concatenation of P and (the end of) A_{n-1} is an Ariadne thread for W.

Proposition 3.1 *Every extendable 2-dimensional SAW can be generated by our algorithm.*

Proof. We only have to prove that, in case (iii) of the algorithm above, the path P contains points at L_∞-distance 1 from w_{n-1}.

In this case, up to symmetries and rotations, we are in one of the cases of Figure 3. Since W is extendable, in all cases, the points marked with a white circle cannot be on W_{n-1}: if a point w_i ($i < n-1$) is one of these points, $(w_i, \ldots, w_{n-2}, w_{n-1})$ is a loop which separates the plane into two connected components, one containing w_n and the other one containing a_0. By Jordan's Theorem, one of these two components is finite, which is impossible since both a_0 and w_n are "linked to the infinite" by a walk which does not cross W_{n-1}.

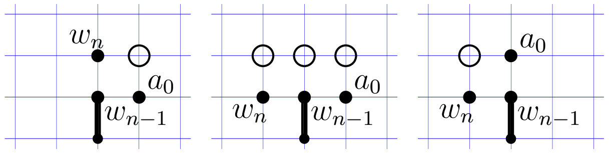

Fig. 3. The three cases with the point w_n which does not see the infinite and is not on A_{n-1}.

So, by adding (in the right order) the points marked with a white circle at the beginning of A_{n-1}, we get an Ariadne thread for W (it is possible that one of these white points is already in A_{n-1}, in which case we only have to suppress the loop inside A). □

Proposition 3.2 *If we assume that the basic operations on a hash table (i.e., searching, adding or deleting an element) run in time $O(1)$, then adding a new point to an extendable SAW runs in time $O(1)$. Thus, generating a extendable SAW of length n in $\mathbb{Z}^d$ takes $O(n)$ time.*

Proof. We represent the walk W by a list W^L (which will be the output) and $d+1$ associative arrays:

- W_P, whose keys are the points, and which has no values.

- For each coordinate i, an associative array W_i whose keys are the $d-1$ remaining coordinates of the points and such that the value $W_i[(x_1, x_2, \dots x_{i-1}, x_{i+1}, \dots, x_d)]$ associated with a key $(x_1, x_2, \dots x_{i-1}, x_{i+1}, \dots, x_d)$ is given by $[y_{min}, y_{max}]$, where y_{min} (resp. y_{max}) is the smallest (resp. greatest) ith coordinate of points in W having $x_1, \dots x_{i-1}$, $x_{i+1}, \dots, x_d$ as other coordinates (if there is no such point in W, the key $(x_1, x_2, \dots x_{i-1}, x_{i+1}, \dots, x_d)$ does not exist).

When we create a new thread (i.e. when we add to the walk a point that sees the infinite), we only store the starting point of the half-line. We represent the Ariadne thread A by a (finite) list A^L (the first element of A^L is the oldest in the thread) and an associative array A_P. The keys of A_P are the points of A; if a is a point of A, $A_P[a]$ is the index of a in A^L.

In order to add a new point to W, we consider (in random order) the neighbors of the last point w_{n-1} of W. Let $w = (x_1, \dots x_d)$ be one of these neighbors, then we perform the following actions (in the order they are listed):

(i) check if $w \in W$;

(ii) check if w sees the infinite (and modify W and A if this is the case);

(iii) check if $w \in A$.

(iv) check if there is a path not crossing W between w and A contained in the neighborood of w_{n-1}.

In order to check if $w \in W$, we just have to test if w is a key of W_p. This takes $O(1)$.

To check if w sees the infinite, we have to compare each coordinate x_i with the values y_{min} and y_{max} stored in $W_i[(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)]$ (w sees the infinite if and only if there exists i such that $x_i < y_{min}$ or $x_i > y_{max}$). This test takes $O(1)$. If w sees the infinite, then we have to add it in W^L and W_p; we also have to modify the W_i (if $x_i < y_{min}$, $y_{min} \leftarrow x_i$ and if $x_i > y_{max}$, $y_{max} \leftarrow x_i$) and to create A^L and A_P with one point inside. All these actions takes time $O(1)$.

Checking if $w \in A$ takes $O(1)$. If this is the case, we have to add w to W and also to delete all points of A which have been added (in A) after w, i.e. all the elements of A^L with index $\geq A_P[(x, y)]$ (with all the corresponding elements of A_P). Deleting one point takes $O(1)$, but it is possible that, at one step, many points of A have to be deleted. At each step, we add at most $2^d - 1$ points in A. So, between the beginning of the generation of W and Step n, it is impossible to delete more than $(2^d - 1)n$ elements of A. The amortized complexity of this step is thus $O(1)$.

For the last test, we only have to find a path in a graph with at most $2^d - 1$ vertices and to add one point to W and at most $2^d - 1$ points to A. This takes $O(1)$. We may have to delete a loop in A but, similarly to the previous case, the amortized complexity of this deletion is $O(1)$. □

Proposition 3.3 *Generating an extendable SAW of length n in $\mathbb{Z}^2$ takes $O(n)$ time.*

Proof. It is sufficient to replace, in the proof of Proposition 3.2, the associative arrays W_P, W_i (for $i = 1, 2$) and A_P by data structures whose access time is $O(1)$ in the worst case. The associative arrays W_P and A_P can be replaced by the data structure of [4], as adapted to $\mathbb{Z} \times \mathbb{Z}$ in [2]. Each W_i can be replaced by two lists (one for the positive coordinates, and the other one for the negative coordinates). Since the paths start from the origin, if a point (x, y) is in a path, there will be a point with abscissa $x - 1$ if $x > 0$ or $x + 1$ if $x < 0$ (and similarly for ordinate y). So these lists will have no unused cases. □

Remark. It is possible (for the general algorithm in $\mathbb{Z}^n$) to use balanced search trees instead of associative arrays. In this case, the complexity of adding a point to an extendable SAW is $O(\log n)$ and so the complexity of generating an extendable SAW of length n is $O(n \log n)$.

In dimension 2, our algorithm is very close to the one in [3,5] which relies on a "winding number" that cannot be generalized in higher dimensions. The advantage of our algorithm is that it works in any dimension. In dimension > 2, our algorithm generates a subset of the extendable SAWs. We conjecture that the set of the SAWs generated by our algorithm is large enough to admit a connective constant equal to the one for general SAWs (hints for this results are given by the fact that extendable SAWs that cannot be generated by our algorithm have to contain very particular configurations and that for dimension 3 and larger the estimated critical exponent ν of our extendable SAWs equals the one obtained for random walks (see below)).

3.2 Tests and variants

Since the algorithm is linear, it is possible to generate long SAWs in rather short time. Table 1 shows results of tests made in Python 2.7 on an Intel Core i5 at 2.7 GHz. Although these tests were made in $\mathbb{Z} \times \mathbb{Z}$, we used the algorithm with associative arrays. It is possible to generate, in approximatively

12 seconds, an extendable SAW of length 10^6. We obtain SAWs like those in Figure 4.

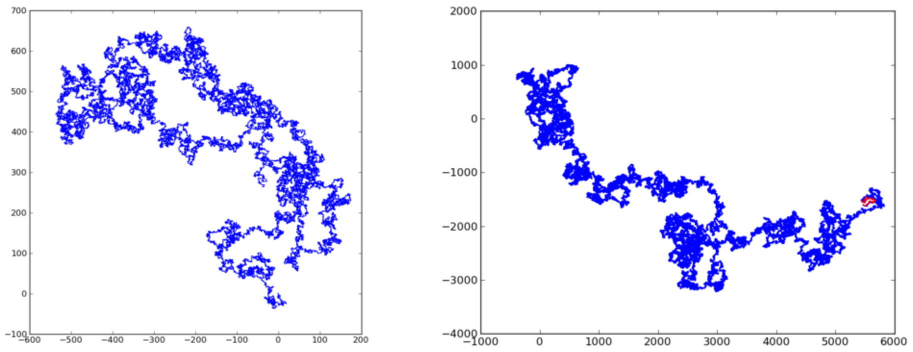

Fig. 4. Two SAWs on $\mathbb{Z} \times \mathbb{Z}$ obtained with our algorithm. One of length 50000 and one of length 10^6.

length	total time for 100 trials (s)	average time
1000	0.920	9 ms
5000	4.885	49 ms
10000	9.879	99 ms
50000	52.569	526 ms
100000	106.056	1.06 s
500000	570.968	5.7 s
1000000	1175.337	11.8 s

Table 1
Time taken for the generation of extendable SAWs.

We have measured the average distance between the two extremities of extendable SAWs. We have made 20000 trials on SAWs of length 100, 200, . . . , 900, 1000, 2000,. . . , 10000, 20000, . . . until 10^6. The results are shown in Figure 5.

The slope of the straight line of Figure 5 is 0.571, and its correlation coefficient is > 0.999. From these tests, we got .57 as an estimated value for the critical exponent ν for the lattice $\mathbb{Z} \times \mathbb{Z}$, which confirms the value got in [13] by enumeration of all the extendable SAWs of length ≤ 22. This value

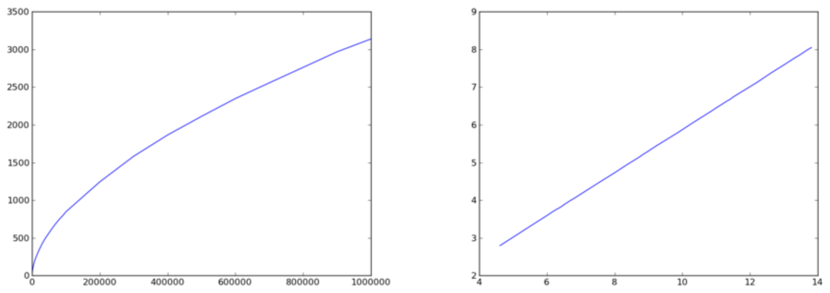

Fig. 5. Average distance between the two extremities of extendable SAWs on $\mathbb{Z}\times\mathbb{Z}$, for lengths in $\{100, 200, \dots, 1000, 2000, \dots, 10^4, \dots 10^6\}$. The curve on the left represents the average distance as a function of the length. The curve on the right represents the log of the average distance as a function of the log of the length.

is different from the one expected for the uniform distribution (.75); actually, our algorithm generates a kinetic distribution [16] of extendable SAWs.

Similarly, we got $\nu \approx .51$ for the lattice $\mathbb{Z}^3$ and $\nu \approx .50$ for $\mathbb{Z}^4$, $\mathbb{Z}^5$ and $\mathbb{Z}^6$. These results are coherent with the fact that in high dimensional space, SAW and random walks are quite similar.

We can change the distribution of SAWs generated by our algorithm in the following way: in the lattice $\mathbb{Z} \times \mathbb{Z}$, let $x_1, \dots x_4$ be the neighbors of the last point of the SAW. If it is possible to extend W with x_i, we define $p_i = K + h_i$, where h_i is the number of half-lines starting at x_i and not crossing W, and K is a fixed parameter. If it is not possible to extend W with x_i, define $p_i = 0$. Then we choose x_i with probability $p_i / \sum_{j=1}^{4} p_j$. Such a modification disadvantages dense SAWs. On the contrary, by choosing $p_i = K + 4 - h_i$, we favor dense SAWs. In Figure 6, one can see some SAWs that we get , with different corrections. These different appearances correspond to different values of the critical exponent ν (see Figure 8).

We have chosen a "local" correction because we want to avoid backtracking. So this variant of our algorithm is also linear in time; it takes approximatively twice the time of the original one. Another (a posteriori) reason is that "it works", in the sense that, for every value of K, we can get a value for the critical exponent ν. As for the basic algorithm, this value is the slope of a straight line (the log of the average distance as a function of the log of the length) and, in each case, the correlation coefficient is > 0.997.

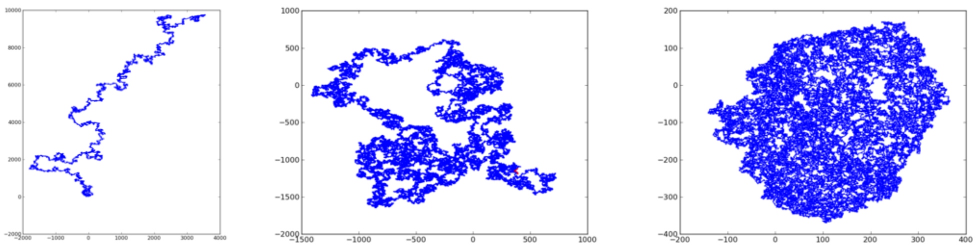

Fig. 6. 2-dimensional SAWs of length $5 \cdot 10^5$, with correction $p_i = 7 + h_i$, with no correction and with correction $p_i = 4 - h_i$ (from left to right).

3.3 Adaptation of the algorithm to other tilings

The definition of the Ariadne thread does not depend on the tiling. The only configuration that changes is the one were the last point of the walk does not see the infinite and is not on the thread. Figure 7 shows some cases for the hexagonal and the triangular lattice in the plane.

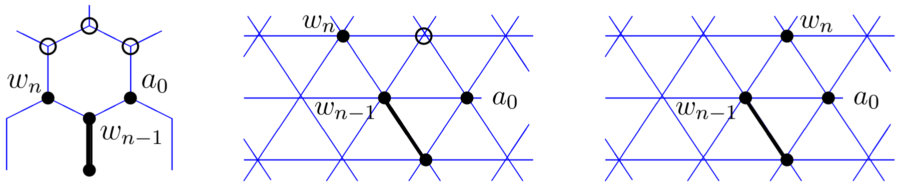

Fig. 7. Configurations where the last point of an extendable SAW does not see the infinite and is not on the Ariadne thread for the hexagonal and the triangular lattices. For the triangular lattice, it is possible that w_n is a neighbor of a_0 (on the right of the figure); there is only one such configuration (up to rotations and symmetries). There are three configurations like the one of the middle of the figure, where w_n is not a neighbor of a_0. Notice that, in all cases, a_0 is a neighbor of the two last points of the walk. For the hexagonal lattice, there is only one configuration.

4 Conclusion

We have presented in this paper a linear, and thus optimal, algorithm to generate random self avoiding walks on the lattice $\mathbb{Z}^d$. This algorithm generates extendable SAWs, and these SAWs have the same connective constant as the general SAWs. Our algorithm produces a kinetic distribution on the extendable SAWs for which the critical exponent ν is ≈ 0.57 in dimension 2 (.51 if $d = 3$ and .50 for $d = 4$, 5 or 6). We have also presented a vari-

1	.966	.490
2	.917	.492
3	.869	.496
4	.827	.499
5	.793	.504
6	.763	.507
7	.740	.511
8	.721	.514
9	.704	.518
10	.691	.520
11	.678	.521
12	.669	.524
13	.661	.526
14	.654	.528
15	.648	.529
16	.643	.531
17	.638	.533
18	.633	.534
19	.630	.535
20	.627	.536
22	.622	.538
25	.615	.542
30	.607	.545
40	.598	.550
100	.580	.562

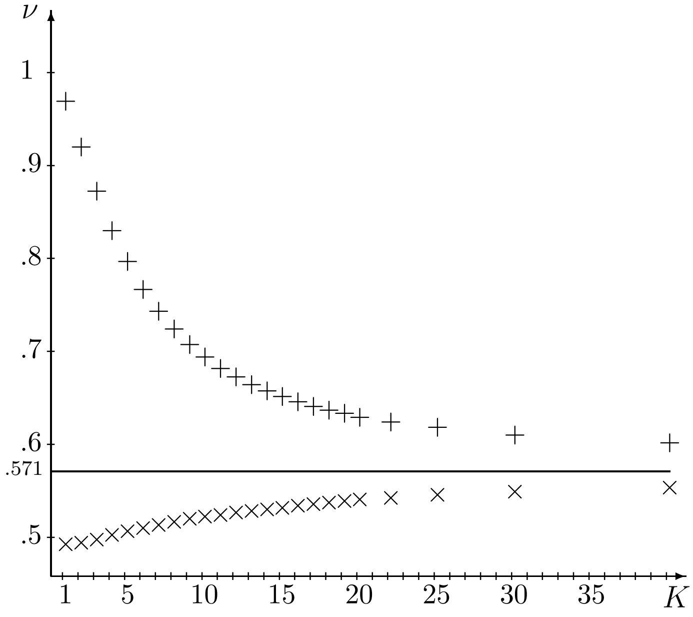

Fig. 8. Critical Exponent ν depending on the parameter K. In the table, K is in the first column, ν for correction $p_i = K + h_i$ is in the second column, and ν for correction $p_i = 4+K-h_i$ is in the third columns. The x-axis of the figure represents K, and ν is on the y-axis. The values for $p_i = K + h_i$ are marked with $+$, and those for $p_i = 4 + K - h_i$ with $\times$. All these values have been estimated with 2000 trials on SAWs of length $100, 200, \ldots, 900, 1000, 2000 \ldots 9000, 10000, \ldots 10^6$. In all cases, the correlation coefficient of the log-log line is $> .997$.

ant of our algorithm, also linear, which depends on a parameter K. For any $x \in [0.49, \ldots, 0.966]$, it is possible to determine a value of K such that our algorithm gives a distribution of self-avoiding walks with critical exponent $\nu = x$.

Our algorithm can easily be adapted to other lattices like the Archimedean lattices of the plane since, in every tiling of the plane, the construction /modification of an Ariadne thread yields the same three cases. It is also possible to "prune-enrich" it [9].

References

[1] Berretti, A. and A. D. Sokal, *New Monte Carlo Methods for the Self-Avoiding Walk*, J. Stat. Phys. **40** (1985), pp. 483–531.

[2] Blondin Massé, A. and É. Marcotte, *A generic data structure for representing discrete paths on regular grids*, GASCom 2016, Bastia.

[3] Bousquet-Mélou, M., *On the Importance Sampling of Self-Avoiding Walks*, Combin. Probab. Comput. **23** (2014), pp. 725–748.

[4] Brlek, S., M. Koskas and X. Provençal, *A linear time and space algorithm for detecting path intersections*, Theoret. Comput. Sci. **412** (2011), pp. 4841–4850.

[5] Chan, Y. B. and A. Rechnitzer, *A Monte-Carlo Study of Non-Trapped Self-Avoiding Walks*, J. Phys. A **45** (2012), 405004.

[6] Duminil-Copin, H. and S. Smirnov, *The Connective Constant of the Honeycomb Lattice Equals* $\sqrt{2+\sqrt{2}}$, Ann. of Math. **175** (2012), pp. 1653–1665.

[7] Flory, P. J., *The Configuration of a Real Polymer Chain*, J. Chem. Phys. **17** (1949), pp. 303–310.

[8] Gennes, P. G. de, *Exponents for the Excluded Volume Problem as Derived by the Wilson Method*, Phys. Lett. A **38** (1972), pp. 339–340.

[9] Grassberger, P., *Pruned-Enriched Rosenbluth Method: Simulation of θ-Polymers of Chain Length up to 1000000*, Phys. Rev. E (3) **56** (1997), pp. 3682–3693.

[10] Grimmett, G. R., A. E. Holroyd and Y. Peres, *Extendable Self-Avoiding Walks*, Ann. Inst. Henri Poincaré Comb. Phys. Interact., **1** (2014), pp. 61–75.

[11] Jensen, I. and A. J. Guttmann, *Self-Avoiding Polygons on the Square Lattice*, J. Phys. A **32** (1999), pp. 4867–4876.

[12] Kennedy, T., *A Faster Implementation of the Pivot Algorithm for Self-Avoiding Walks*, J. Stat. Phys. **106** (2002), pp. 407–429.

[13] Kremer, K. and J. W. Lyklema, *Indefinitely growing self-avoiding walk*, Phys. Rev. Lett. **54** (1985), pp. 267–269.

[14] Lal, M., *Monte Carlo Computer Simulation of Chain Molecules*, Mol. Phys. **17** (1969), pp. 57–64.

[15] Madras, N. and G. Slade, “The Self-Avoiding Walk”, Birkhäuser, 1993.

[16] Majid, I., N. Jan, A. Coniglio and H. E. Stanley, *Kinetic Growth Walk: a New Model for Linear Polymers*, Phys. Rev. Lett. **52** (1984), pp. 1257–1260.

[17] Metropolis, N. and S. Ulam, *The Monte-Carlo Method*, J. Amer. Statist. Assoc. **44** (1949), pp. 335–341.

[18] Rechnitzer, A. and E. J. Janse van Rensburg, *Generalized Atmospheric Rosenbluth Methods (GARM)*, J. Phys. A **41** (2008), pp. 442002-442010.

[19] Rosenbluth, M. N. and A. W. Rosenbluth, *Monte Carlo Calculations of the Average Extension of Molecular Chains*, J. Chem. Phys. **36** (1955), pp. 356–359.