The study of unfoldable self-avoiding walks. Application to protein structure prediction software

Christophe Guyeux[a], Jean-Marc Nicod[a], Laurent Philippe[a,*], Jacques M. Bahi[a]

[a] *FEMTO-ST Institute, UMR 6174 CNRS, University of Franche-Comté, Besançon, France*

Abstract

Self-avoiding walks (SAWs) are the source of very difficult problems in probability and enumerative combinatorics. They are of great interest as, for example, they are the basis of protein structure prediction in bioinformatics. The authors of this article have previously shown that, depending on the prediction algorithm, the sets of obtained walk conformations differ: for example, all the self-avoiding walks can be generated using stretching-based algorithms whereas only the unfoldable SAWs can be obtained with methods that iteratively fold the straight line. A deeper study of (non-)unfoldable self-avoiding walks is presented in this article. The contribution is first a survey of what is currently known about these sets. In particular we provide clear definitions of various subsets of self-avoiding walks related to pivot moves (unfoldable and non-unfoldable SAWs, etc.) and the first results we have obtained, theoretically or computationally, on these sets. Then a new theorem on the number of non-unfoldable SAWs is demonstrated. Finally, a list of open questions is provided and the consequences on the protein structure prediction problem is proposed.

Keywords: Protein structure prediction, Protein folding, Self-avoiding walks, Combinatorics algorithms, Problem complexity, Discrete structures.

1. Introduction

Self-avoiding walks (SAWs) have been studied over decades, both for their interest in mathematics and their applications in physics: standard model of long chain polymers [13], fundamental example in the theory of critical phenomena in equilibrium statistical mechanics [26, 11], and so on. They are the source of very difficult problems in probability and enumerative combinatorics [2, 7], regarding among other things the number of $n-$step SAWs, their mean-square

[*] Corresponding author

Email addresses: `christophe.guyeux@femto-st.fr` (Christophe Guyeux), `jean-marc.nicod@femto-st.fr` (Jean-Marc Nicod), `laurent.philippe@femto-st.fr` (Laurent Philippe), `jacques.bahi@femto-st.fr` (Jacques M. Bahi)

displacement, and the so-called scaling limit. The self-avoiding walks naturally appear in bioinformatics during the prediction of the 3D conformation of a protein of interest. Frequently, the two dimensional backbone of the protein is looked for in a first stage, and then this 2D structure is refined step by step to obtain the final 3D conformation.

Protein Structure Prediction (PSP) software can be separated into various categories. We decide in this article to focus on the two following frequently used ones. On the one hand, some algorithms construct the proteins' structures on the 2D or 3D square lattice by adding, at each iteration, a new amino acid at the tail of the protein. Most of the time, various positions are possible for this amino acid, and the chosen position is the one that optimizes a given functional, for instance the number of neighboring hydrophobic amino acids. On the other hand, some algorithms start from the straight line with the size of the considered protein, and they iterate pivot moves on this structure. Pivot amino acids and angles are chosen to optimize a well-defined energy function. We have pointed out, in our previous researches on the dynamics of the protein folding process [5, 4], that these two categories of protein structure prediction software cannot produce the same conformations [14]. More precisely all the conformations can be attained in the first category whereas it is not the case in the second one. This result has been formerly discovered by the community of mathematicians that studies the self-avoiding walks (SAWs). It seems however to be ignored by bioinformaticians and the connection with the PSP problem has not been signaled.

Indeed, in their article introducing the pivot algorithm [22], Madras and Sokal demonstrate a theorem showing that, when starting from the straight line of length n, and iterating the 180° rotation and either both 90° rotations or both diagonal reflections, all the $n-$step self-avoiding walks on $\mathbb{Z}^2$ can be obtained. In other words, their pivot algorithm is ergodic for this set of transformations. As an example, they depicted in this article a 223-step SAW in $\mathbb{Z}^2$ that is not connected to any other SAW by 90° rotations (see Figure 1). This first apparition of a "non-unfoldable" SAW was indeed the unique one in the literature, and the study of (non-)unfoldable SAWs has not been deepened before our work in [14].

Thus, by using the pivot algorithm incorrectly, a class of walks is excluded. A first appropriate response to this issue is obviously to correct the software so that the pivot algorithm is used correctly. If these walks however constitute an exponentially small subset of SAWs, the lack of ergodicity in existing software might not be fatal, and the results in the biology literature produced using these tools may remain correct. Thus, additionally to the intrinsic theoretical interest to study a kind of walks that has not yet been regarded, the determination of the size ratio between self-avoiding walks and non-unfoldable SAWs may impact both the development of new PSP software and some results previously published in the proteomics area. This article does not solve the question of the size ratio but it produces first theoretical framework and results that may help to evaluate it in further studies.

The contribution of this article is thus a list of first results and questionings about various sets of self-avoiding walks that can (or cannot) be attained by ±90° pivot moves, and their consequences regarding the PSP software. After recalling some basis on self-avoiding walks, we provide definitions of 4 subsets of SAWs that appear when considering pivot moves, namely the folded SAWs obtained by iterating pivot moves on the straight line, the non-unfoldable SAWs, the set of SAWs that can be unfolded at least once, and finally the subset of self-avoiding walks that can be folded k times, $k > 1$. A list of results on these subsets is provided. Among other things, the cardinality of unfoldable SAWs has been bounded, the existence of infinitely many non-unfoldable SAWs has been proven, a shorter example of non-unfoldable walk is given (107 steps), whereas the equality between the set of SAWs and the set of unfoldable SAWs has been computationally verified until number of steps lower or equal to 14. Relations between these subsets are also provided before listing various open problems on (non-)unfoldable self-avoiding walks. Computational aspects of this study are detailed in [6].

The remainder of this document is organized as follows. In the next section, a short overview about the self-avoiding walks is provided. This section enables us to introduce basic definitions and well-known results concerning these walks. Section 3 contains the rigorous definition of the subsets of self-avoiding walks regarded in this manuscript. Then, in Section 4, a first list of easy-to-obtain results we have obtained concerning the subset of non-unfoldable SAWs is detailed, whereas the main result of this article is proven in the next section. A non-exhaustive list of open questions is drawn up in Section 6. Consequences regarding the protein structure prediction problem are investigated in Section 7. This research work ends by a conclusion section, in which the contributions are summarized and intended future work is proposed.

2. A Short Overview of Self-Avoiding Walks

We first recall usual notations and well-known results regarding self-avoiding walks. In a next section, we bring partially these results in the unfoldable SAWs subset.

2.1. Definitions and Terminologies

Let $\mathbb{N}$ be the set of all natural numbers, $\mathbb{N}^* = \{1, 2, \ldots\}$ the set of all positive integers, and for $a, b \in \mathbb{N}$, $a < b$, the notation $[\![a, b]\!]$ stands for the set $\{a, a+1, \ldots, b-1, b\}$. $|x|$ stands for the Euclidean norm of any vector $x \in \mathbb{Z}^d, d \geqslant 1$, whereas $x_1, \ldots, x_d$ are the d coordinates of x. The $n-$th term of a sequence s is denoted by $s(n)$. Finally, $\sharp X$ is the cardinality of a finite set X.

Let us now introduce the notion of self-avoiding walk [23, 26, 18].

Definition 1 (Self-Avoiding Walk) Let $d \geqslant 1$. A $n-$step *self-avoiding walk* from $x \in \mathbb{Z}^d$ to $y \in \mathbb{Z}^d$ is a map $w : [\![0, n]\!] \to \mathbb{Z}^d$ with:

- $w(0) = x$ and $w(n) = y$,
- $|w(i+1) - w(i)| = 1$,

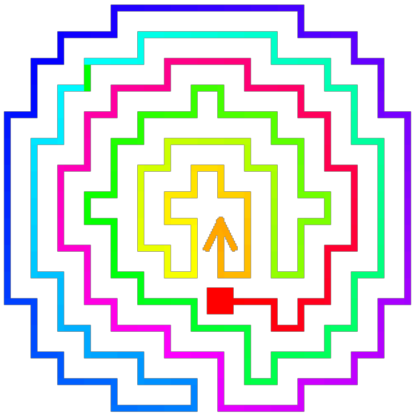


Figure 1: The first SAW shown to be not connected to any other SAW by 90° rotations (Madras and Sokal, [22]), that is, the first discovered non-unfoldable SAW.

- $\forall i, j \in [\![0, n]\!]$, $i \neq j \Rightarrow w(i) \neq w(j)$ (self-avoiding property).

Let $d \in \mathbb{N}^*$. $\mathcal{S}_n(x)$ is the set of $n-$step self-avoiding walks on $\mathbb{Z}^d$ from 0 to x, $c_n(x) = \sharp\mathcal{S}_n(x)$ is the cardinality of this set, $\mathcal{S}_n = \cup_{x\in\mathbb{Z}^d}\mathcal{S}_n(x)$ is constituted by all $n-$step self-avoiding walks that start from 0, whereas $c_n = \sum_{x\in\mathbb{Z}^d} c_n(x)$ is the number of $n-$step self-avoiding walks on $\mathbb{Z}^d$ starting from 0, that is, $c_n = \sharp\mathcal{S}_n$ [26].

2.2. Well-known results about self-avoiding walks

The objective of this section is not to realize a complete state of the art about established or conjectured results on SAWs, but only to present a short list of properties that are connected to our first investigations. For instance the well-known pattern theorem [23] is not presented here. For further results about SAWs readers can consult for instance [26, 23].

A first result concerning the number of $n-$step self-avoiding walks can be easily obtained by remarking that, when $m-$step SAWs are concatenated to $n-$step SAWs, we found all $(m + n)-$step self-avoiding walks *and* other walks having intersections. In other words,

Proposition 1 $\forall m, n \in \mathbb{N}^*, c_{m+n} \leqslant c_m c_n$.

The existence of the so-called *connective constant* is a consequence of such a proposition.

Theorem 1 *The limit* $\lim_{n\to\infty} c_n^{1/n}$ *exists. It is called the* connective constant *and is denoted by* μ. *Moreover, we have* $\mu^n \leqslant c_n$ *and* $d \leqslant \mu \leqslant 2d - 1$.

For a proof of this result, reader is referred to [23].

Various bounds or estimates can be found in the literature [21, 26], like $c_n \approx A\mu^n n^{\gamma-1}$ for A and γ to determine (predicted asymptotic behavior) and

$$\mu \in [2.625662, 2.679193].$$

The pivot algorithm is a dynamic Monte Carlo algorithm that produces self-avoiding walks using the following basic approach [22]. Firstly, a point p on the walk w is picked randomly and used as a pivot. Then a random symmetry operation of the lattice, like a rotation, is applied to the second part (suffixes) of the walk, using p as origin. If the resulting walk is a SAW, it is accepted, else it is rejected and w is counted once again in the sample. A more detailed and precise algorithm can be found in [22]. In this latter article, it is shown that, quoting Madras and Sokal,

Theorem 2 *The pivot algorithm is ergodic for self-avoiding walks on $\mathbb{Z}^d$ provided that all axis reflections, and either all 90° rotations or all diagonal reflections, are given nonzero probability. In fact, any $N-$step SAW can be transformed into a straight rod by some sequence of $2N-1$ or fewer such pivots.*

The pivot algorithm is ergodic too for SAWs on the square lattice [22], provided that the 180° rotation, and either both 90° rotations or both diagonal reflections, are given nonzero probability, whereas 90° rotations alone are not enough, due to Fig. 1.

3. Introducing the (non-)unfoldable self-avoiding walks

3.1. Protein folding as preliminaries

Let us introduce the original context motivating the study of particular subsets of SAWs we called "unfoldable" self-avoiding walks in the remainder of this document.

The 2 or 3 dimensional square lattice hydrophobic-hydrophilic model, simply denoted as *HP model*, is used for low resolution backbone structure prediction of a given protein. In this model formerly introduced by Dill [12], hydrophobic interactions are supposed to dominate protein folding [5, 4]. The protein core freeing up energy is formed by hydrophobic amino acids, whereas hydrophilic amino acids tend to move in the outer surface due to their affinity with the solvent (see Fig. 2).

In this model a protein conformation is a SAW on a 2D or 3D lattice depending on the level of resolution. This SAW depends on topological neighboring contacts between hydrophobic amino acids that are not contiguous in the primary structure. The SAW is such that the free energy E of the protein is minimal. In other words, for an amino acid sequence P of length n and for the set $\mathcal{C}(P)$ of all $n-$step SAWs, the walk chosen to represent the conformation of the protein is $C^* = min\,\{E(C) \mid C \in \mathcal{C}(P)\}$ [25]. In that context and for a conformation (SAW) C, $E(C) = -q$ where q is equal to the number of topological hydrophobic neighbors. For example, $E(c) = -5$ in Fig. 2.

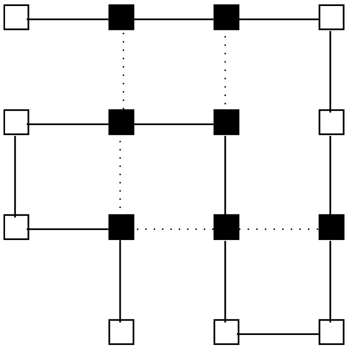


Figure 2: Hydrophilic-hydrophobic model (black squares are hydrophobic residues)

The overriding problem in PSP is: *how to find such a minimal conformation, given all the $n-$step self-avoiding walks and the sequence of hydrophobicity of the protein ?*

Given its sequence of hydrophobicity, finding the best 2D conformation of a protein is not an easy task. When considering the set of self-avoiding walks having $n-$steps and whose vertices are either black (hydrophobic) or white squares (hydrophylic residues), the authors of [10] have indeed proven that determining the SAWs that maximize the number of neighboring black squares in this set is NP-hard. Given a sequence of amino acids, such statement leads to the use of heuristics to predict (and not to determine exactly) the most probable conformation of the protein. These heuristics operate as in the real biological world, folding or increasing the length of SAWs in order to minimize the free energy of the associated conformation. By doing so the protein synthesis in its aqueous environment is reproduced *in silico*. As stated previously, we have shown in a previous work that such investigations potentially lead to various subsets of self-avoiding walks [5, 4, 14].

In the first approach, starting from the straight line, we obtain by a succession of pivot moves of 90° a final conformation being a self-avoiding walk. In this approach, it is not regarded whether the intermediate walks are self-avoiding or not. Such a method corresponds to programs that start from the initial conformation, fold several times the linear protein, according to their embedded scoring functions, and then obtain a final conformation on which the SAW requirement is verified. It is easy to be convinced that, by doing so, the set of final conformations is exactly equal to the set of self-avoiding walks having n steps. As the conformations obtained by such methods coincide exactly to the well-studied global set of all SAWs, such an approach is not further investigated in the remainder of this paper [14].

Remark 1 This first approach guarantees to reach all self-avoiding walks. The embedded energy or scoring function however discriminate against some walks, which is indeed the role of this function. For instance, a straight chain is a SAW but most reasonable scoring functions for folding will not produce this as a

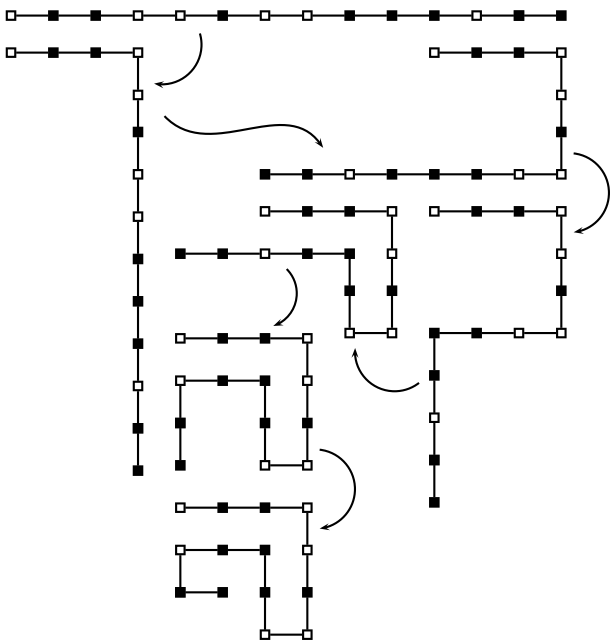


Figure 3: Protein Structure Prediction by folding SAWs

"folded" conformation, as the pivot moves are accepted only when energy/score can be lowered. We simply point out that, in this first approach, the walks that are evaluated by the scoring function is the whole set of self-avoiding walks: simulating numerically protein folding does not introduce a bias before calling the scoring function.

In the second approach, the same process is realized, except that all the intermediate conformations must be self-avoiding walks (see Fig. 3). The set of $n-$step SAWs reachable by such a procedure is denoted by $fSAW_n$ in what follows. Such a procedure is one of the two most usual translations of the so-called “SAW requirement” in the bioinformatics literature, leading to proteins' conformations belonging into $fSAW_n$. For instance, PSP methods presented in [19, 27, 8, 15, 17] follow such an approach. We have shown in [14] that $fSAW_n \subsetneq \mathcal{S}_n$ [22]. In other words, *in this first category of PSP software, it is impossible to reach all the conformations of $\mathcal{S}_n$.*

Remark 2 This second approach discards intermediate conformations with collisions. This may prevent a method from reaching specific SAWs. In particular, protein structure prediction and protein folding are not two sides of the same coin.

Other approaches in the same category can be imagined, like the following one. We can act as above, requiring additionally that no intersection of vertex

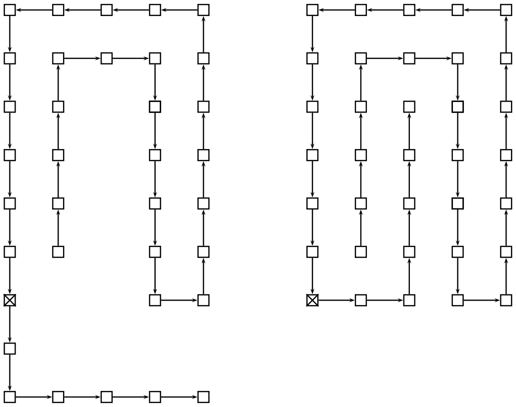


Figure 4: Pivot move acceptable in $fSAW$ but not in $fSAW'$

or edge during the transformation of one SAW to another occurs. For instance, the pivot move of Figure 4 is authorized in the previous $fSAW$ approach, but it is refused in the current one: during the rotation around the residue having a cross, the rigid structure after this residue intersects the remainder of the "protein" (see Fig. 5). In this two dimensional approach denoted by $fSAW'$, it is not allowed for a protein folding to use the 3D space to achieve one plane conformation from another plane one. A reasonable modeling of the true natural folding dynamics of an already synthesized protein can be obtained by extending this requirement to the third dimension. However, due to its complexity, this requirement is actually never used by tools that embed a 2D HP square lattice model for protein structure prediction. This is why these particular SAWs are not further investigated in this document. Let us just emphasize that $fSAW'_n$ is obviously a subset of $fSAW_n$, but there is *a priori* no reason to consider them equal. Indeed, Figure 6 shows that,

Proposition 2 *For all* $n \in \mathbb{N}^*$, $fSAW'_n \subset fSAW_n$. *However,* $\exists n \in \mathbb{N}^*$, $fSAW'_n \neq fSAW_n$.

Proof In Figure 6, the unique possible pivot move is the red dot, and obviously such move leads to the intersection between the head and the tail of the structure during the transformation.

Note that we only studied pivot moves of ±90° in the three previous approaches. But considering other sets of transformations could be interesting in some well-defined contexts and can potentially lead to new subsets of SAWs.

A last bioinformatics approach of protein structure prediction using self-avoiding walks starts with an $1-$step SAW, and at iteration k, a new step is added at the tail of the walk, in such a way that the new $k-$step self-avoiding walk presents the best value for the considered scoring function (see Fig 7). The protein is thus constructed step by step, reaching the best local conformation at

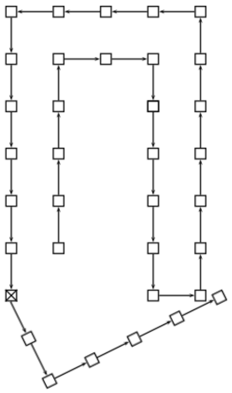


Figure 5: An intersection appears between the head and the tail during the transformation, thus this pivot move is refused in $fSAW'$.

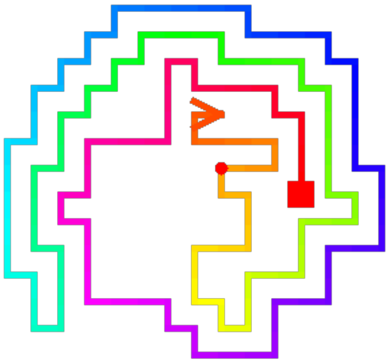


Figure 6: $fSAW_n \neq fSAW'_n$

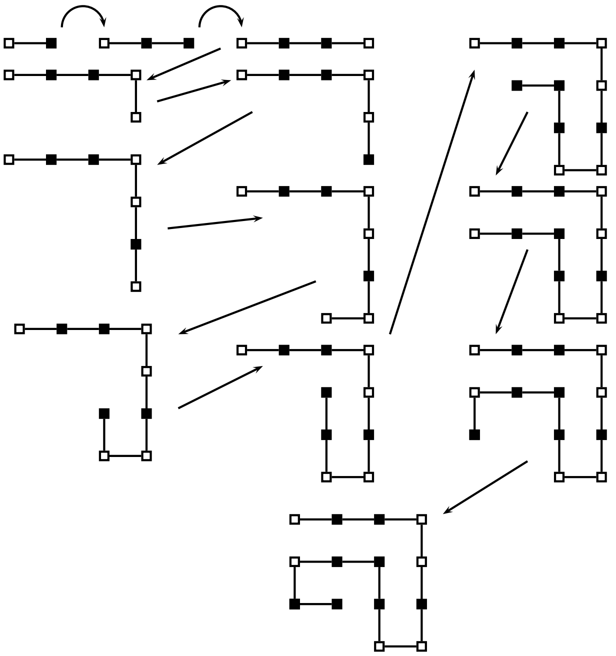


Figure 7: Protein Structure Prediction by stretching SAWs

each iteration. It is easy to see that such an approach leads to all the possible self-avoiding walks having the length of the considered protein [14].

In the remainder of this document, we give a more rigorous definition of the $fSAW_n$ set, we initiate its study, and compare it to the well-known SAWs set denoted $\mathcal{S}_n$.

3.2. Notations

Unfoldable self-avoiding walks can be studied in a lattice having d dimensions. However, for the sake of simplicity, authors of this research work have decided to introduce them only on the 2 dimensional square lattice $\mathbb{Z}^2$, to be as close as possible to their field of application: the low resolution backbone structure prediction of a protein. Such restriction enables us to produce understandable pictures of such not yet investigated particular walks.

One of the easiest way to define the previously described self-avoiding walks that appear during the realization of the SAW requirement in PSP algorithms, is to introduce the absolute encoding of a walk [16, 3]. In this encoding, a $n+1-$step walk $w = w(0), \ldots, w(n) \in \left(\mathbb{Z}^2\right)^{n+1}$ with $w(0) = (0,0)$ is a sequence $s = s(0), \ldots, s(n-1)$ of elements belonging into $\mathbb{Z}/4\mathbb{Z}$, such that:

- $s(i) = 0$ if and only if $w(i+1)_1 = w(i)_1 + 1$ and $w(i+1)_2 = w(i)_2$, that is, $w(i+1)$ is at the East of $w(i)$.

- $s(i) = 1$ if and only if $w(i+1)_1 = w(i)_1$ and $w(i+1)_2 = w(i)_2 - 1$: $w(i+1)$ is at the South of $w(i)$.
- $s(i) = 2$ if and only if $w(i+1)_1 = w(i)_1 - 1$ and $w(i+1)_2 = w(i)_2$, meaning that $w(i+1)$ is at the West of $w(i)$.
- Finally, $s(i) = 3$ if and only if $w(i+1)_1 = w(i)_1$ and $w(i+1)_2 = w(i)_2 + 1$ ($w(i+1)$ is at the North of $w(i)$).

Let us now define the following functions [14].

Definition 2 The *anticlockwise fold function* is the function $f : \mathbb{Z}/4\mathbb{Z} \longrightarrow \mathbb{Z}/4\mathbb{Z}$ defined by $f(x) = x - 1 \pmod 4$ and the clockwise fold function is $f^{-1}(x) = x + 1 \pmod 4$.

Using the absolute encoding sequence s of a $n-$step SAW w that starts from the origin of the square lattice, a pivot move of 90° on $w(k)$, $k < n$, simply consists to transform s into $s(0), \ldots, s(k-1), f(s(k)), \ldots, f(s(n))$. Similarly, a pivot move of $-90°$ consists to apply f^{-1} to the tail of the absolute encoding sequence, like in Figure 8.

3.3. A graph structure for SAWs folding process

We can now introduce a graph structure describing well the iterations of ±90° pivot moves on a given self-avoiding walk.

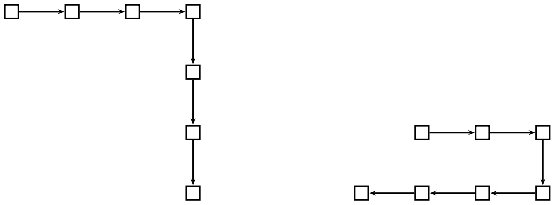
Given $n \in \mathbb{N}^*$, the graph $\mathfrak{G}_n$, formerly introduced in [14], is defined as follows:

- its vertices are the $n-$step self-avoiding walks, described in absolute encoding;
- there is an edge between two vertices s_i, s_j if and only if s_j can be obtained by one pivot move of ±90° on s_i, that is, if there exists $k \in [\![0, n-1]\!]$ s.t.:
  - either $s_i(0), \ldots, s_i(k-1), f(s_i(k)), \ldots, f(s_i(n)) = s_j$
  - or $s_i(0), \ldots, s_i(k-1), f^{-1}(s_i(k)), \ldots, f^{-1}(s_i(n)) = s_j$.

Such a digraph is depicted in Figure 9. The circled vertex is the straight line whereas strikeout vertices are walks that are not self-avoiding. Depending on the context, and for the sake of simplicity, $\mathfrak{G}_n$ will also refer to the set of SAWs in $\mathfrak{G}_n$ (*i.e.*, its vertices).

Using this graph, the unfoldable SAWs introduced in the previous section can be redefined more rigorously.

Definition 3 $fSAW_n$ is the connected component of the straight line $00\ldots0$ (n times) in $\mathfrak{G}_n$, whereas $\mathcal{S}_n$ is constituted by all the vertices of $\mathfrak{G}_n$.



(a) 000111

(b) 001222 $= 00f^{-1}(0)f^{-1}(1)f^{-1}(1)f^{-1}(1)$

Figure 8: Effects of the clockwise fold function applied on the four last components of an absolute encoding.

The Figure 1 shows that the connected component $fSAW(223)$ of the straight line in $\mathfrak{S}_{223}$ is not equal to the whole graph: $\mathfrak{S}_{223}$ is not connected. More precisely, this graph has a connected component of size 1: Figure 1 is totally non-unfoldable, whereas SAW of Fig. 6 can be folded exactly once. Indeed, to be in the same connected component is an equivalence relation $\mathcal{R}_n$ on $\mathfrak{S}_n, \forall n \in \mathbb{N}^*$, and two SAWs w, w' are considered equivalent (with respect to this equivalence relation) if and only if there is a way to fold w into w' such that all the intermediate walks are self-avoiding. When existing, such a way is not necessarily unique.

These remarks lead to the following definitions.

Definition 4 Let $n \in \mathbb{N}^*$ and $w \in \mathcal{S}_n$. We say that:

- w is *non-unfoldable* if its equivalence class, with respect to $\mathcal{R}_n$, is of size 1;
- *w is an unfoldable self-avoiding walk* if its equivalence class contains the $n-$step straight walk $000\ldots0$ ($n-1$ times);
- *w can be folded k times* if a simple path of length k exists between w and another vertex in the same connected component of w.

Moreover, we introduce the following sets:

- $fSAW(n)$ is the equivalence class of the $n-$step straight walk, or the set of all unfoldable SAWs.
- $fSAW(n,k)$ is the set of equivalence classes of size k in $(\mathfrak{S}_n, \mathcal{R}_n)$.
- $USAW(n)$ is the set of equivalence classes of size 1 $(\mathfrak{S}_n, \mathcal{R}_n)$, that is, the set of non-unfoldable walks.
- $f^1SAW(n)$ is the complement of $USAW(n)$ in $\mathfrak{S}_n$. This is the set of SAWs on which we can apply at least one pivot move of $\pm 90°$.

Example 1 Figure 10 shows the two elements of a class belonging into $fSAW(219,2)$ whereas Fig. 1 is an element of $USAW(223)$.

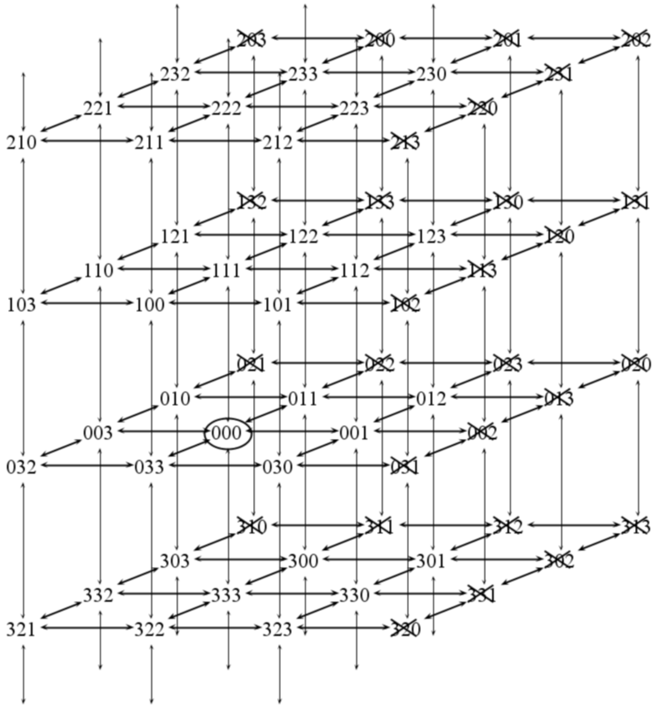


Figure 9: The digraph $\mathfrak{G}_3 = fSAW(3)$

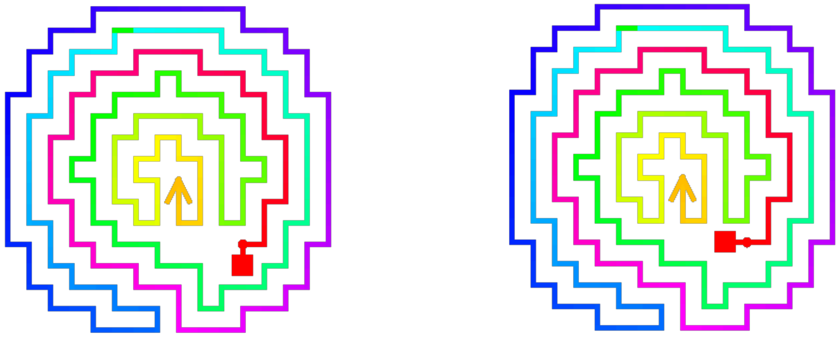


Figure 10: The two self-avoiding walks in $fSAW(219, 2)$

4. A Short List of Results on (non-)unfoldable Self-Avoiding Walks

We now give a first collection of easy-to-obtained results concerning the particular SAW sets introduced in the previous section. These results have been either obtained mathematically or by using computers.

We firstly show that,

Proposition 3 *The cardinality ϕ_n of $fSAW_n$ satisfies: $2^{n+2} \leqslant \phi_n \leqslant 4 \times 3^n$.*

This result is a consequence of the following lemma.

Lemma 1 *The 2^n $n-$step walks that take steps only in a set of coordinate directions having the form $\{i, i+1 (mod\ 4)\}$ are in $fSAW(n)$.*

This lemma can be proven using the number of cranks of a self-avoiding walk, defined below.

Definition 5 (Crank) Let w be a $n-$step self-avoiding walk on $\mathbb{Z}^2$ of absolute encoding s. w contains a crank at position $k \in [\![1,n]\!]$ if $s(k-1) \neq s(k)$.

Proof (Lemma 1) Let $n \in \mathbb{N}^*$. We show by a mathematical induction that, $\forall N \in \mathbb{N}$, any $n-$step self-avoiding walk that (1) has coordinate directions belonging in a set of the form $\{i, i+1 (mod\ 4)\}$, for a given $i \in [\![0,3]\!]$, and (2) has N cranks, is in $fSAW(n)$.

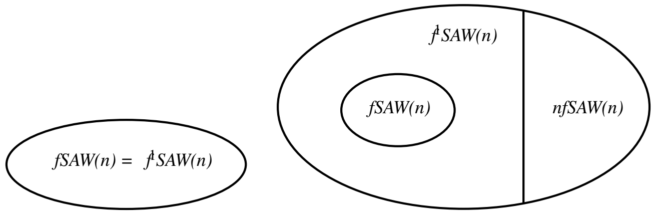
The base case is obvious, as if $N = 0$, then w is a straight line.

Let $N \in \mathbb{N}$ such that the statement holds for all $k \leqslant N$, and consider a $n-$step self-avoiding walk w that has $N+1$ cranks while taking steps only in the positive coordinate directions (set of coordinate directions having the form $\{0,3\}$), in order to clarify expectations. Let j be the position of the first crank in w. As steps are taken only in the positive coordinate directions, only two situations can occur: (1) $w(j) = w(j-1) + (1,0)$ and $w(j+1) = w(j) + (0,1)$ ($s(j-1) = 0, s(j) = 3$), or (2) $w(j) = w(j-1) + (0,1)$ and $w(j+1) = w(j) + (1,0)$ ($s(j-1) = 3, s(j) = 0$).

Suppose now that the origin of the 2D square lattice is set to $w(j)$. So, in the first situation (1),

- $\forall l > j$, $w(l) = (w(l)_1, w(l)_2)$ is such that $w(l)_1 \geqslant 0$ while $w(l)_2 \geqslant 1$,
- $\forall l < j$, $w(l) = (w(l)_1, w(l)_2)$ is such that $w(l)_1 \leqslant -1$ while $w(l)_2 \leqslant 0$.

The effect of a 90° pivot move on the origin $w(j)$ is to reduce the number of cranks $N+1$ to N in w, and to map each $w(l) = (w(l)_1, w(l)_2)$ into $(w(l)_2, w(l)_1)$, $\forall l > j$. After such a pivot move, the obtained walk w' is such that $\forall l > j$, $w'(l)_1 = w(l)_2 \geqslant 1$, while $\forall l < j$, $w'(l)_1 = w(l)_1 \leqslant -1$. In other words, the walk w' still remains self-avoiding. The induction hypothesis is then applied on the tail of w' (the head being the $j-1$-step straight line), which satisfies the two required properties (w' having N cranks, its tail has $N-1$ cranks). Furthermore, w' is obtained by operating a pivot move on w, thus these two walks belong into



(a) $\mathfrak{G}_n$ for $n \leqslant 14$ (b) Diagram of $\mathfrak{G}_n$ for $n = 107$

Figure 11: Vien diagram for $\mathfrak{G}_n$

the same connective component of $\mathfrak{G}_n$. Finally, $w \in fSAW(n)$. This argument still remains valid for the other sets $\{i, i+1 (mod\, 4)\}, i \in [\![0, 2]\!]$, *mutatis mutandis.*

The second situation (2) can also be handled that way, which concludes the mathematical induction and the proof of the lemma.

Proof (Proposition 3) Due to Lemma 1, we have $\phi_n \geqslant 4 \times 2^n$ (4× because of the 4 quarters of the square lattice). And since the set of $n-$step walks without immediate reversals has cardinality 4×3^n and contains all $n-$step unfoldable self-avoiding walks, we have $\phi_n \leqslant 4 \times 3^n$.

Remark 3 SAWs whose absolute encoding is only constituted by 0's and 1's are unfoldable SAWs. It is also possible that a few 2's or 3's can be added without breaking the unfoldable character of the walk. This means that the lower bound could be increased.

Proposition 4 *$\forall n \leqslant 14, fSAW(n) = \mathfrak{G}_n$ whereas $fSAW(107) \subsetneq \mathfrak{G}_{107}$ (see Figure 11).*
In other words, let ν_n the smallest $n \geqslant 2$ such that $USAW(n) \neq \emptyset$. Then $15 \leqslant \nu_n \leqslant 107$.

Proof We have realized a program that constructs the connected component of the $n-$step straight line for $n \leqslant 14$, and at each time, we have obtained the whole $\mathfrak{G}_n$ (see [6]). Additionally, using a backtracking method, we have obtained the walk depicted in Figure 12. This walk justifies the upper bound of 107: we have verified using a systematic program that no pivot move can be realized in that walk without breaking the self-avoiding requirement. These programs, their explanations and justifications can be found in [6].

Proposition 5 $\forall n \leqslant 28, f^1SAW(n) = \mathfrak{G}_n$.

Proof Obtained experimentally, see [6].

The results contained into the two previous propositions are summarized, with all intermediate computations, in Table 1. The $\sharp\mathfrak{G}_n$ values, obtained in [20], are recalled here for comparison.

n	$\sharp\mathfrak{G}_n$	$\sharp f^1SAW(n)$	$\sharp USAW(n) =$ $\sharp\overline{f^1SAW(n)}$	$\sharp fSAW(n)$
1	4	4	0	4
2	12	12	0	12
3	36	36	0	36
4	100	100	0	100
5	284	284	0	284
6	780	780	0	780
7	2172	2172	0	2172
8	5916	5916	0	5916
9	16268	16268	0	16268
10	44100	44100	0	44100
11	120292	120292	0	120292
12	324932	324932	0	324932
13	881500	881500	0	881500
14	2374444	2374444	0	2374444
15	6416596	6416596	0	?
16	17245332	17245332	0	?
17	46466676	46466676	0	?
18	124658732	124658732	0	?
19	335116620	335116620	0	?
20	897697164	897697164	0	?
21	2408806028	2408806028	0	?
22	6444560484	6444560484	0	?
23	17266613812	17266613812	0	?
24	46146397316	46146397316	0	?
25	123481354908	123481354908	0	?
26	329712786220	329712786220	0	?
27	881317491628	881317491628	0	?
28	2351378582244	2351378582244	0	?
29	6279396229332	?	?	?
30	16741957935348	?	?	?
31	44673816630956	?	?	?
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
107	?	?	$\geqslant 1$	?

Table 1: Cardinalities of various subsets of SAWs

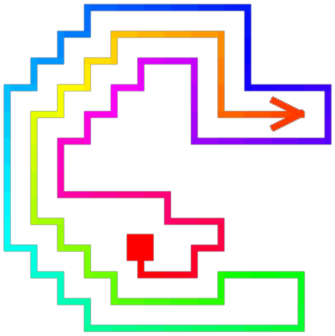


Figure 12: Current smallest (107-step) SAW that cannot be folded

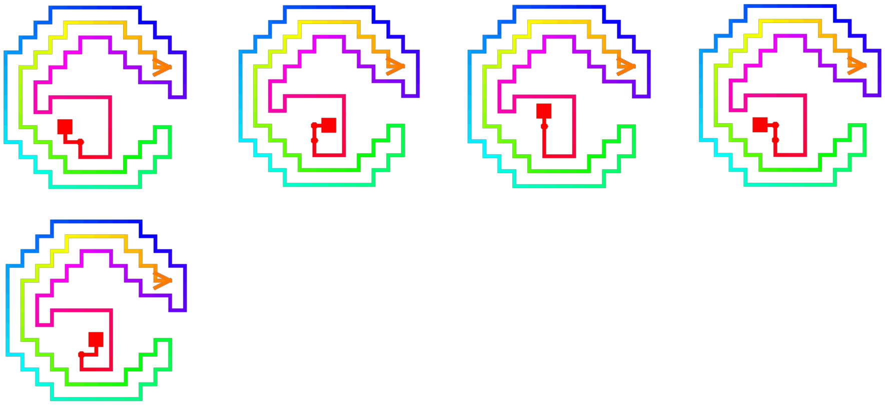


Figure 13: A connected component with 5 elements

Until now, connected components presented in this paper either have the straight line, or are of size 1 or 2. A reasonable questioning is to wonder whether it is possible to have larger connected components different from the one of the straight line. We are founded to claim that,

Proposition 6 *It exists $k > 2$ such that $fSAW(n,k)$ is nonempty.*

In other words, connected components different from $fSAW(n)$ and larger than 1 or 2 elements exist. The result, which has been experimentally obtained, can be proven by exhibiting a example: Figure 13 shows a connected component of size 5.

We can define a diameter function D on the connected components of $\mathfrak{G}_n$, such that $D(C)$ is the length of the longest shortest path in the connected component C of $\mathfrak{G}_n$. Consider the connected component of the straight line $fSAW(n)$, we have the result,

Proposition 7 *The diameter of $fSAW(n)$ is equal to $2n$: $D(fSAW(n)) = 2n$.*

PROOF We take the SAW S_{z_1} defined as the zigzag $(0,1,0,1,0,...)$ and the S_{z_2} defined as the zigzag $(2,1,2,1,2,...)$.

We can transform S_{z_1} in $(2,3,2,3,2,...)$ by two pivot moves:

$$(0,1,0,1...) \to (1,2,1,2,1,...) \to (2,3,2,3,2,...).$$

Then two other pivot moves allow us to transform $(2,3,2,3,2,...)$ in $(2,1,0,1,0,...)$, that is,

$$(2,3,2,3,2,...) \to (2,2,1,2,1,2,...) \to (2,1,0,1,0,1,...).$$

As the respective visited vertices start by $(0,1),(1,2),(2,3),(2,2),(2,1)$, we obtain by doing so a simple path of length 4. The process can be reproduced on the tail $(0,1,0...)$ of $(2,1,0,1,0...)$ until each 0's (odd positions) of the SAW has been transformed to 2, and each 1's (even position) has been set again to 1. As there are two pivot moves for each value in the path and as each pivot move is in a different direction in $\mathfrak{G}_n$, so the minimum distance from S_{z_1} to S_{z_2} in $\mathfrak{G}_n$ is $2n$.

This path, from S_{z_1} to S_{z_2}, is the largest distance we can find in $\mathfrak{G}_n$ as we have two pivot moves on each edge. If we add indeed one more pivot move, i.e., three pivot moves, on an edge then the same value could be obtained from the initial position by making only one pivot move in the opposite direction which would reduce the distance between the two SAWs.

Example 2 In $fSAW(2)$, this diameter corresponds, for instance, to the shortest path $03 \to 00 \to 11 \to 12 \to 23$ (see Figure 14).

5. The Main Results

Let us introduce again new notations and terminologies.

5.1. Definitions and properties

Definition 6 Let $x(n)$ be the word

$03^{4n}(23)^4 2^{8n+2}(12)^4 1^{8n+2}(01)^4 0^{8n+2}(30)^4 3^{4n} 01^{4n+1}(21)^4 2^{8n+4}(32)^4 3^{8n+4}(03)^4 0^{8n+4}(10)^4 1^{4n+2} 2^2$,

of length $64n+89$ and $y(n)$ be the word

$0^2 1^{4n+3}(21)^4 2^{8(n+1)}(32)^4 3^{8(n+1)}(03)^4 0^{8(n+1)}(10)^4 1^{4n+4} 23^{4n+3}(23)^4 2^{8n+6}(12)^4 1^{8n+6}(01)^4 0^{8n+6}(30)^4 3^{4n+1} 2$.

of length $64n+121$. Let s_0 be the word:

2,1,2,2,3,2,3,2,3,3,0,3,0,3,0,0,1,0,1,0,0,3,2,3,2,3,2,3,2,2,1,2,1,2,1,2,1,2,1,1,0,1,0,
1,0,1,0,1,0,0,3,0,3,0,3,0,3,0,0,0,0,1,1,1,2,1,2,1,2,1,2,1,2,2,2,2,2,2,2,2,3,2,3,2,3,2,
3,2,3,3,3,3,3,3,3,3,0,3,0,3,0,3,0,3,0,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,1,1,1,2,3,3,3,2,
3,2,3,2,3,2,3,2,2,2,2,2,2,1,2,1,2,1,2,1,2,1,1,1,1,1,1,0,1,0,1,0,1,0,1,0,0,0,0,0,0,3,0,
3,0,3,0,3,0,3,2,2,1,2,1,2,1,2,1,2,2,2,2,3,2,3,2,3,2,3,2,3,3,3,3,0,3,0,3,0,3,0,3,0,0,0,
0,1,0,1,0,1,0,1,0,1,1,2,2,2,2,2,3,2,2,1,1,0,0,0,

whose walk $w_0 = dec((2,-2);s_0)$ is depicted in Figure 15(a). $s_k, k \geqslant 1$, is inductively defined by:

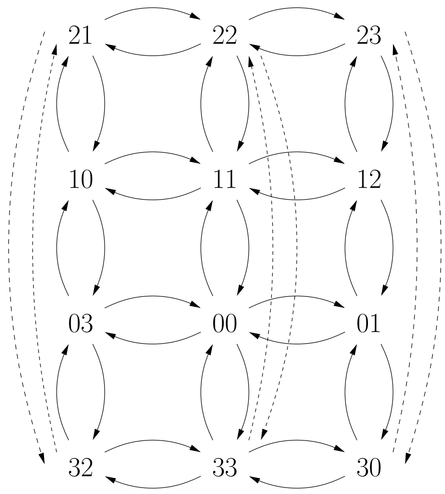


Figure 14: The digraph $\mathfrak{G}_2 = fSAW(2)$

- if k is even, s_k is s_{k-1} in which the word $1^{4n-1}123^{4n-1}$ has been replaced by $1^{4n-1}x(n)23^{4n-1}$, where $n = k/2 + 1$,
- otherwise, when k is odd, s_k is s_{k-1} in which $3^{4n}011^{4n}$ is substituted by $3^{4n}0y(n)1^{4n}$, where n is $(k+1)/2$.

For $n > 1$, we define the walk w_n by: $w_n = dec((2, -2); s_n)$.

This process is well defined, as an immediate recursion shows that for all $n > 0$:

- s_{2n} contains the subword $1^{4n}23^{4n-1}$ exactly once and it does not contain the subword $3^{4n}01^{4n+1}$, while $x(n)$ introduces this latter word exactly once into s_{2n}.
- s_{2n+1} contains the subword $3^{4n}01^{4n+1}$ exactly once. It has no subword equal to $1^{4(n+1)}23^{4(n+1)-1}$, whereas $y(n)$ introduces it in s_{2n+1} exactly once.

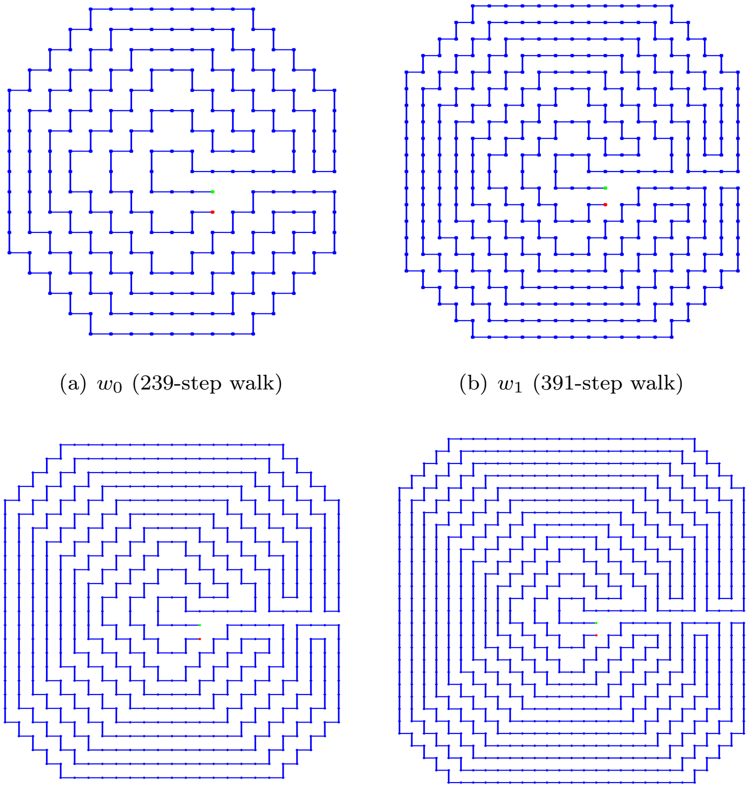
The word s_{2n} can thus be divided in three parts: the pattern $1^{4n-1}123^{4n-1}$, its left part σ^l_{2n} and its right part σ^r_{2n}. A similar notation can be introduced for the three equivalent subword in s_{2n+1}.

This recursive process is illustrated in Fig. 23. We can claim that,

Theorem 3 *For all n, $w_n = dec((2, -2); s_n)$ is in $nfSAW(|w_n|)$.*

Remark 4 w_0 is a 239-step walk, whereas for all $n > 0$:

- if n is odd, then $|w_n| = |w_{n-1}| + |y(\frac{n-1}{2})| - 1 = |w_{n-1}| + 64 \times \frac{n-1}{2} + 120$,
- else $|w_n| = |w_{n-1}| + |x(\frac{n}{2} + 1)| - 1 = |w_{n-1}| + 64 \times (\frac{n}{2} + 1) + 88$.



(a) w_0 (239-step walk)

(b) w_1 (391-step walk)

(c) w_2 (575-step walk)

(d) w_3 (791-step walk)

Figure 15: Generating walks that cannot be folded

See Figure 23 for a representation of $w_0, \ldots, w_3$ and Table 2 for the first sizes of w_n.

Corollary 1 *There is an infinite number of n such that $nfSAW(n)$ is nonempty. In particular, the number of SAWs that cannot be folded is infinite.*

Proof (Corollary) This is an immediate consequence of Theorem 3, as $|w_n|$ is a strictly increasing sequence. □

5.2. Proof of Theorem 3

5.2.1. Preliminaries

Consider the octagons O_n and o_n on the square lattice, respectively bounded by $\{\mathsf{A}_n, \mathsf{B}_n, \mathsf{C}_n, \mathsf{D}_n, \mathsf{E}_n, \mathsf{F}_n, \mathsf{G}_n, \mathsf{H}_n, \mathsf{I}_n, \mathsf{J}_n\}$ and by $\{\mathsf{a}_n, \mathsf{b}_n, \mathsf{c}_n, \mathsf{d}_n, \mathsf{e}_n, \mathsf{f}_n, \mathsf{g}_n, \mathsf{h}_n, \mathsf{i}_n, \mathsf{j}_n\}$, where:

n	$\|w_n\|$	n	$\|w_n\|$
0	239	10	3199
1	391	11	3671
2	575	12	4175
3	791	13	4711
4	1039	14	5279
5	1319	15	5879
6	1631	16	6511
7	1975	17	7175
8	2351	18	7871
9	2759	19	8599

Table 2: Size of w_n

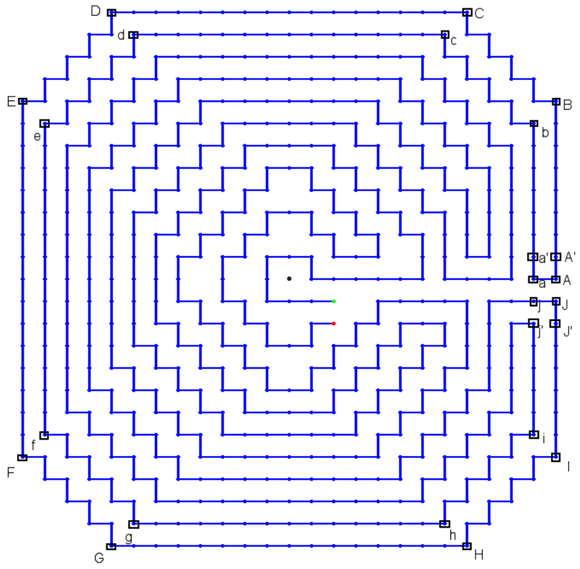


Figure 16: Remarkable points of octagons o_{n-1} and O_{n-1} defining walk o_{n-1} in w_n ($n = 2$ here).

- $\mathsf{a}_n = (2n+9, 0), \mathsf{A}_n = (2n+10, 0)$,
- $\mathsf{b}_n = (2n+9, 2n+5), \mathsf{B}_n = (2n+10, 2n+6)$,
- $\mathsf{c}_n = (2n+5, 2n+9), \mathsf{C}_n = (2n+6, 2n+10)$,
- $\mathsf{d}_n = (-2n-5, 2n+9), \mathsf{D}_n = (-2n-6, 2n+10)$,
- $\mathsf{e}_n = (-2n-9, 2n+5), \mathsf{E}_n = (-2n-10, 2n+6)$,
- $\mathsf{f}_n = (-2n-9, -2n-5), \mathsf{F}_n = (-2n-10, -2n-6)$,
- $\mathsf{g}_n = (-2n-5, -2n-9), \mathsf{G}_n = (-2n-6, -2n-10)$,
- $\mathsf{h}_n = (2n+5, -2n-9), \mathsf{H}_n = (2n+6, -2n-10)$,
- $\mathsf{i}_n = (2n+9, -2n-5), \mathsf{I}_n = (2n+10, -2n-6)$,
- and $\mathsf{j}_n = (2n+9, -1), \mathsf{J}_n = (2n+10, -1)$.

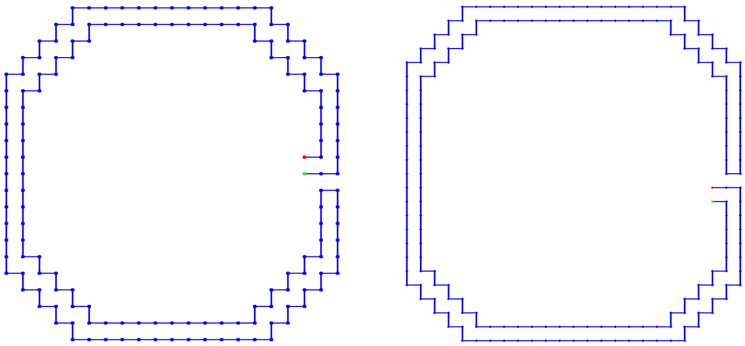
Let $\mathsf{a}'_n = (2n+9, 1)$, $\mathsf{A}'_n = (2n+10, 1)$, $\mathsf{j}'_n = (2n+9, -2)$, and $\mathsf{J}'_n = (2n+10, -2)$, as depicted in Figure 16, while $\mathsf{x}_n = (2n+8, 1)$, $\mathsf{X}_n = (2n+8, 0)$, $\mathsf{y}_n = (2n+8, -2)$, and $\mathsf{Y}_n = (2n+8, -1)$ are at the West of these points.

For $n \in \mathbb{N}$, define the walk o as follows: if n is even, then $o_n = dec(\mathsf{x}_n, x(n/2+1))$, else $o_n = dec(\mathsf{Y}_n, y((n+1)/2))$. In other words, o is alternatively x and y, depending on the parity of n (o_0 and o_1 are depicted in Figure 17). Having the encoding of o_n, it is immediate to prove that, when n is even, then o_n is the walk that:

1. starts from x_n,
2. moves one step in the East until reaching a'_n,
3. then visits the points b_n, $\mathsf{c}_n, \mathsf{d}_n$, $\mathsf{e}_n, \mathsf{f}_n$, $\mathsf{g}_n, \mathsf{h}_n, \mathsf{i}_n, \mathsf{j}_n$ of o_n in the anticlockwise direction, by starting with a_n and following a straight line (or a diagonal),
4. moves another step in the East direction,
5. then visits all the points of discrete octagon O_n in clockwise direction, that is, J_n, I_n, H_n, G_n, F_n, E_n, D_n, C_n, B_n, and finally A_n,
6. terminates its move by two steps in the West direction, until reaching X_n.

When n is odd, as o_1 in Figure 17, we obtain a similar move, *mutatis mutandis*: octagon O_n is visited first (always in the clockwise direction), x_n is replaced by Y_n, and a'_n by j_n. Remark that, knowing the lengths of $x(n)$ and $y(n)$, we can check that o_n is a $32n+153$-step walk.

Graphically speaking, the recursive process presented in Definition 6 corresponds to starts with the walk w_0, and to add recursively o_n in w_n to obtain w_{n+1} (see Figure 18). This insertion occurs between x_n and X_n if n is even, and between Y_n and y_n else, as proven by the following lemma.



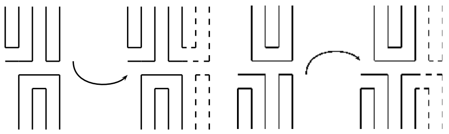
(a) o_0 (153-step walk)

(b) o_1 (185-step walk)

Figure 17: First o_n walks

Lemma 2 *Let $s_{2n} = s^l_{2n}1s^r_{2n}$, where $s^l_{2n} = \sigma^l_{2n}1^{4n-1}$ and $s^r_{2n} = 23^{4n-1}\sigma^r_{2n}$. Then the point $w_{|s^l_{2n}|+1} \in \mathbb{Z}^2$, where the extension of w_{2n} begins during its transformation in $w_{2n+1} = enc(s^l_{2n}x(n)s^r_{2n})$, is x_{2n}. The end of the extension $w_{|s^l_{2n}|+|x(n)|+|s^r_{2n}|+1}$, for its part, is in X_{2n}.*

Similarly, the extension of the walk w_{2n+1} from $enc(s_{2n+1})$ to $enc(s_{2n+2})$ starts in Y_{2n+1} and it ends in y_{2n+1}.



(a) Case 1

(b) Case 2

Figure 18: Opening walks

Proof This lemma can be proven by induction on n.

The base case is obvious, as w_1 is obtained by opening and extending w_0 between $(8,1) = \mathsf{x}_0$ and $(8,0) = \mathsf{X}_0$, whereas w_2 is obtained by extending w_1 between $(10,-1) = \mathsf{Y}_1$ and $(10,-2) = \mathsf{y}_1$.

Suppose now that $n > 0$. On the one hand, let us consider $s_{2n} = s^l_{2n}1s^r_{2n}$. We have:

$$\begin{aligned} s_{2n+1} &= s^l_{2n}x(n)s^r_{2n} \\ &= s^l_{2n}03^{4n}(23)^4 2^{8n+2}(12)^4 1^{8n+2}(01)^4 0^{8n+2}(30)^4 3^{4n}0\underline{1}1^{4n} \\ &\quad (21)^4 2^{8n+2}(32)^4 3^{8n+4}(03)^4 0^{8n+4}(10)^4 1^{4n+2}2^2 s^r_{2n} \\ &= s^l_{2n+1}\underline{1}s^r_{2n+1}. \end{aligned}$$

Then the point that starts the extension in $enc(s_{n+1})$ can be obtained from the point that have started the extension in $enc(s_n)$ by following the walk

$03^{4n}(23)^4 2^{8n+2}(12)^4 1^{8n+2}(01)^4 0^{8n+2}(30)^4 3^{4n}0$ before the underlined 1, which will be replaced in the next step by $y(n)$. As:

- the number of 1's in this pattern is $8n+10$ when the number of 3's is $8n+8$,
- and the number of 0's is $8n+12$ while the number of 2's is only $8n+10$,

then this walk corresponds to a move of absolute encoding 0011, that is, a move from x_{2n} (induction hypothesis) to Y_{2n+1}. On the other hand,

$$\begin{aligned} s_{2n+2} = &\quad s^l_{2n+1} y(n) s^r_{2n+1} \\ = &\quad s^l_{2n+1} 0^2 1^{4n+3}(21)^4 2^{8(n+1)}(32)^4 3^{8(n+1)}(03)^4 0^{8(n+1)}(10)^4 1^{4n+3}\underline{1} s^r_{2n+2}. \end{aligned}$$

A similar argument shows that the pattern

$$0^2 1^{4n+3}(21)^4 2^{8(n+1)}(32)^4 3^{8(n+1)}(03)^4 0^{8(n+1)}(10)^4 1^{4n+3}$$

corresponds to a move of absolute encoding 0033, mapping Y_{2n+1} in x_{2n+2}.

Finally, as the number of 0's is equal to the number of 2's in $x(n)$, whereas the number of 1's is the number of 3's plus 1, we have that the additional walk $x(n)$ that has started from x_n ends in X_n.

A similar statement holds for Y_n. □

Example 3 o_0 is inserted between x_0 and X_0 to transform w_0 in w_1. Then o_1 is inserted between Y_1 and y_1 to transform w_1 in w_2, whereas o_2 is inserted between x_2 and X_2 to obtain w_2 from w_1.

Lemma 3 *For all n, the walk w_n is strictly included into the octagon o_n.*

Proof This lemma can be proven by induction on n, as the list of points of w_0 are inside o_0, whereas the induction property comes from the graphical interpretation of Lemma 2. □

We can finally establish the following lemma.

Lemma 4 *All the w_n walks satisfy the self-avoiding property.*

Proof The base case is obvious by construction, as depicted in Figure 15(a). The self-avoiding property of this walk has also been verified computationally, see [6]. The induction is a direct consequence of the previous lemma: the extension o_n is self-avoiding by construction, it is contained into octagons o_n and O_n, whereas the remainder of $w(n+1)$, which is $w(n)$, is self-avoiding due to the inductive hypothesis, and strictly inside o_n due to the previous lemma. □

5.2.2. Proof of the theorem

We can now prove that, for all n, w_n is a non unfoldable self-avoiding walk. The self-avoiding property has been established in Lemma 4, it still remains to demonstrate, by an inductive proof, that $\forall n$, w_n cannot be folded.

The base case, for $n \in \{0,1\}$, has been verified computationally, by testing all the possible pivot moves in w_0 and w_1, and verifying that they are all in contradiction with the self-avoiding property (reader interested by the computational aspects of this work is referred to [6]).

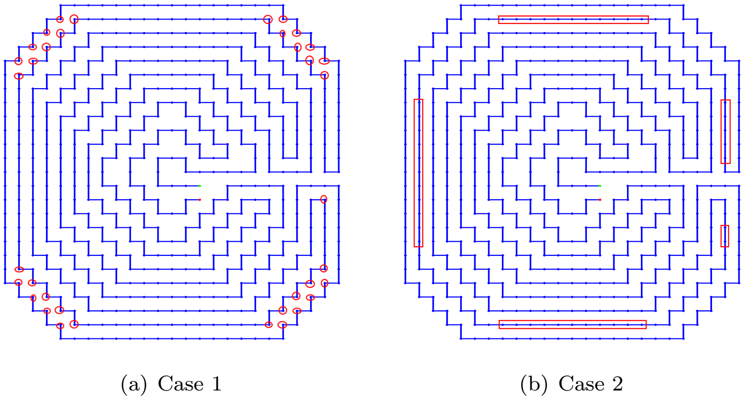
Let $n \geqslant 2$ such that, for all $k < n$, w_k cannot be folded. We will show that w_n cannot be folded too. By construction, w_n is constituted by two subwalks: w_{n-1} and o_{n-1}, as established in the first lemma of this document. As w_{n-1} cannot be folded (this is the inductive hypothesis), we just have to verify that pivot moves on points of the two octagons o_{n-1} and O_{n-1} always lead to a walk that does not satisfy this self-avoiding property.

Suppose now that n is even. As all the points of o_{n-1} are visited in the anticlockwise direction before visiting the points of O_{n-1} in the clockwise one, we have the following result:

- When a point $w_n(i)$ on O_{n-1} has a successor $w_n(i+1)$ at its South (resp. West, North, East), then it has an ancestor $w_n(j), j < i$ immediately at its West (resp. North, East, South). This ancestor is $w_n(i-1)$ when considering the zigzag corners of O_{n-1} whereas it is on o_{n-1} when $w_n(i)$ is strictly inside the up, left, bottom, or right side of O_{n-1}, see Figure 17. A pivot move of $-\pi$ on $w_n(i)$ thus sends $w_n(i+1)$ on $w_n(j)$.

- A similar statement holds for pivot moves of $-\pi$ on points of o_{n-1}.

Such an argument allow us to tackle half of the possible pivot moves, that is, all possible pivot moves of angle $-\pi$.


(a) Case 1

(b) Case 2

Figure 19: Stages of the proof of Theorem 3

Let us now recall that w_{n-1} has been constructed by removing the 1-step segment between x_{n-2} and X_{n-2} in w_{n-2}, and connecting the walk o_{n-2} in this liberated place. Thus w_n contains at least octagons $\mathsf{O}_{n-1}, \mathsf{o}_{n-1}, \mathsf{O}_{n-2}$, and o_{n-2}. And, due to the fact that o_i is visited before O_i when i is even, whereas the situation is reversed when i is odd, we thus have the following travel in w_n:

1. o_{n-2} is firstly visited in anticlockwise direction, from $\mathsf{a}'_{n-2} = (2(n-2)+9, 1)$ until $\mathsf{j}_{n-2} = (2(n-2)+9, -1)$,
2. O_{n-1} is then visited in clockwise direction, from $\mathsf{J}_{n-1} = (2(n-1)+10, -1)$ until reaching $\mathsf{A}_{n-1} = (2(n-1)+10, 0)$,
3. o_{n-1} is then visited in anticlockwise direction, from $\mathsf{a}_{n-1} = (2(n-1)+9, 0)$ to $\mathsf{j}'_{n-1} = (2(n-1)+9, -2)$,
4. finally, O_{n-2} is visited in clockwise direction, from $\mathsf{J}'_{n-2} = (2(n-2)+10, -2)$ to $\mathsf{A}_{n-2} = (2(n-2)+10, 0)$.

On the 4 zigzag corners, points $w_n(i)$ that are such that the absolute encoding of the 2-step walk $w_n(i-1), w_n(i), w_n(i+1)$ is in $\{03, 32, 21, 10\}$, are such that a pivot move of π on $w_n(i)$ maps $w_n(i+1)$ on $w_n(i-1)$. By doing so, we show that all the circled points in Figure 19 cannot be folded without contradicting the self-avoiding property.

Firstly, let us remark that a π pivot move of a_{n-1} sends A_{n-2} at its West on j_{n-1} at its South, and so w_n cannot be folded at this position a_{n-1}. Similarly, j_{n-1} sends a_{n-1} on J_{n-2}, J_{n-1} maps A_{n-1} on j'_{n-1}, and a π pivot move of A_{n-1} sends a_{n-1} on J_{n-1}.

Consider now a π pivot move of a point on the upper side of o_{n-1}, that is, a point $p = (t, 2n+7)$ between $\mathsf{c}_{n-1} = (2n+3, 2n+7)$ and $\mathsf{d}_{n-1} = (-2n-3, 2n+7)$ ($t \in [\![-2n-3, 2n+3]\!]$). Such a pivot move does not modify points of the segment delimited by $\mathsf{c}_{n-2} = (2n+1, 2n+5)$ and $\mathsf{d}_{n-2} = (-2n+1, 2n+5)$, as this part of the octagon o_{n-2} is visited before the upper side of o_{n-1}. Contrarily, all the points between p and d_{n-1} are moved by a pivot move of p, as o_{n-1} is visited in anticlockwise direction. If t is in $[-2n-1, 2n+1]$, then a pivot move of p sends the point in position $p+(-2,0) \in [\mathsf{c}_{n-1}, \mathsf{d}_{n-1}]$ in position $p+(0,-2)$, which is in $[\mathsf{c}_{n-2}, \mathsf{d}_{n-2}]$. As this segment has not been rotated during this pivot move, we thus obtain a contradiction of the self-avoiding property.

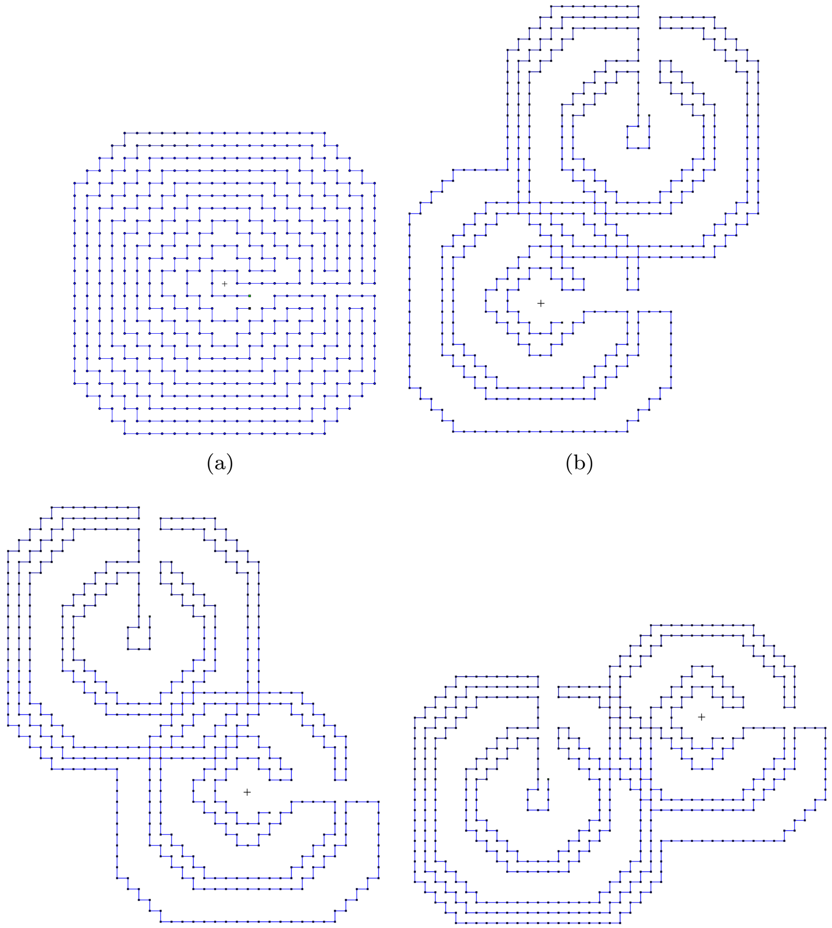
This argument still remains valid for points in the segments $[\mathsf{e}_{n-1}+(0,-2), \mathsf{f}_{n-1}+(0,+2)]$, $[\mathsf{g}_{n-1}+(2,0), \mathsf{h}_{n-1}+(-2,0)]$, $[\mathsf{i}_{n-1}+(0,2), \mathsf{j}'_{n-1}+(0,-2)]$, and $[\mathsf{a}'_{n-1}, \mathsf{b}_{n-1}+(0,-2)]$.

Recall now that a rotation of π centered in (x_0, y_0) maps the point of coordinate (a, b) in the point

$$r_{(a,b)}(x_0, y_0) = (y_0 - b + a, a - x_0 + b). \tag{1}$$

Consider now the points of the upper right internal zigzag between b_{n-1} and c_{n-1}, namely $(2n+6, 2n+3)$, $(2n+5, 2n+4)$, $(2n+4, 2n+5)$, and $(2n+3, 2n+6)$. Remind that O_{n-2} is not affected by pivot moves on o_{n-1}. We can thus verify that:

- a π pivot move of $(2n+6, 2n+3)$ sends $(2n+5, 2n+5) \in \mathsf{o}_{n+1}$ on $(2n+4, 2n+2)$ that belongs in the upper left zigzag Z^{ul}_{n-2} of O_{n-2},
- a π pivot move of $(2n+5, 2n+4)$ sends $(2n+4, 2n+6) \in \mathsf{o}_{n+1}$ on $(2n+3, 2n+3) \in Z^{ul}_{n-2}$,



(a)

(b)

(c)

(d)

Figure 20: Case where n is even (n=2)

- a π pivot move of $(2n+4, 2n+5)$ sends $(2n+3, 2n+7) \in \mathsf{o}_{n+1}$ on $(2n+2, 2n+4) \in Z^{ul}_{n-2}$,
- finally, a π pivot move of $(2n+3, 2n+6)$ sends $(2n+1, 2n+7) \in \mathsf{o}_{n+1}$ on $(2n+2, 2n+4) \in Z^{ul}_{n-2}$ too.

The twelve other points in the zigzags of o_{n-1} can be treated in the same manner, *mutatis mutandis*.

As an extension, the point of coordinate $(2n+7, 2n+2)$ at the South of b_{n-1} maps $(2n+6, 2n+4)$ on $(2n+5, 2n+1) \in \mathsf{O}_{n-2}$, $(-2n-2, 2n+7)$ at the East of d_{n-1} sends $(-2n-4, 2n+6)$ on $(-2n-1, 2n+5) \in \mathsf{O}_{n-2}$, $(-2n-7, -2n-2)$ (at the North of f_{n-1}) sends $(-2n-6, -2n-4)$ on $(-2n-5, -2n-1) \in \mathsf{O}_{n-2}$, while $(2n+2,-2n-7)$ (at the West of h_{n-1}) maps $(-2n-6,-2n-4)$ on $(-2n-5,-2n-1) \in \mathsf{O}_{n-2}$.

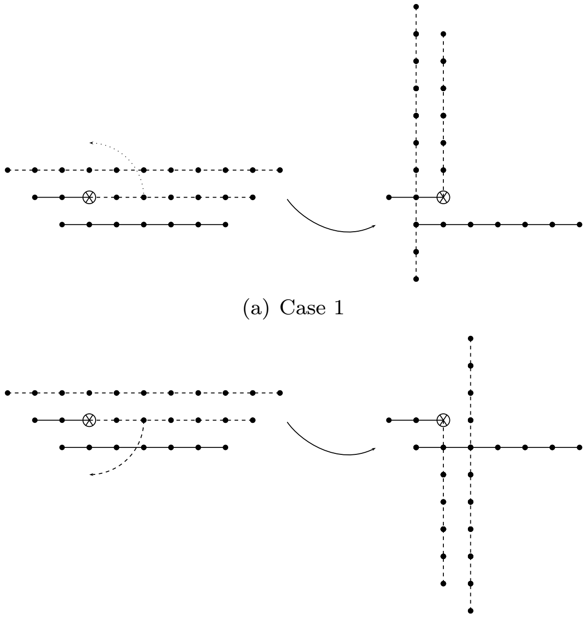
Then a pivot move of $(2n+2,2n+7)$ transforms $(2n-1,2n+7)$ in $(2n+2,2n+4) \in \mathsf{O}_{n-2}$, a pivot move of $(-2n-7,2n+2)$ maps $(-2n-7,2n-1)$ in $(-2n-4,2n+2)$, and a pivot move of $(-2n-2,-2n-7)$ sends $(-2n+1,-2n-7)$ in $(-2n-2,-2n-4)$, while a pivot move of $(2n+7,-2n-2)$ sends $(2n+7,-2n+1)$ in $(2n+4,-2n-2) \in \mathsf{O}_{n-2}$. Finally, a pivot move of $(2n+7,-2n+1)$ maps $(2n+6,0)$ on $(2n+4,-2n) \in \mathsf{O}_{n-2}$, which concludes the study of o_{n-1}, in which no pivot move can be realized without breaking the self-avoiding property of the walk.

Let us now consider the remainder points of O_{n-1}. On the upper side of O_{n-1}, we have the following result: a π pivot move of p between $(-2n,2n+8)$ (4 steps at the West of D_{n-1}) and $(2n,2n+8)$ (4 steps at the East of D_{n-1}). As depicted in Figure 20, and using both Equation 1 and the itinerary sequence stated in the octagons visit list, we can prove that pivot move of p sends the point $p+(-3,-1) \in \mathsf{o}_{n-1}$ in $p+(1,-3)$ belonging in o_{n-2}, leading to a contradiction to the self-avoiding property. As for o_{n-1}, a same statement holds for segments $[(-2n-8,2n),(-2n-8,-2n)]$, $[(-2n,-2n-8);(2n,-2n-8)]$, $[(2n+8,2n),(2n+8,2)]$, and $[(2n+8,-5),(2n+8,-2n)]$, respectively at the West, South, and East sides of octagon O_{n-1}, see Figure 20. Pivot moves on the 4 points at the left of $[(-2n,2n+8);(2n,2n+8)]$ (upper side of O_{n-1}) map the upper left zigzag of o_{n-1} on the upper side of o_{n-2}, whereas the 4 points at its right send the upper side of o_{n-1} on the upper left zigzag of o_{n-2}. Such a statement holds for the four points at the South of E_{n-1} including E_{n-1}, for the 4 points at the North of F_{n-1}, the 4 ones at the East of G_{n-1}, the 4 ones at the West of H_{n-1}, the four points at the North of I_{n-1}, and finally the 4 points at the South of B_{n-1} including B_{n-1}.

The 3 last pivot moves to consider in the external part of the upper right zigzag of O_{n-1} map the upper segment of o_{n-1} in the right segment of o_{n-2}. Similarly, the remainder points of the upper left zigzag map the left side of o_{n-1} on the upper side of o_{n-2}, a pivot move of points in the lower left zigzag of O_{n-1} sends the lower side of o_{n-1} in the left side of o_{n-2}, while the effects of a pivot move of points in the lower right zigzag sends the right side of o_{n-1} on the lower side of o_{n-2}.

It still remains to consider 4 pivot moves. A'_{n-1} sends the point $(2n+6,1)$ on J_{n-1}, while J'_{n-1} and $\mathsf{J}'_{n-1}+(0,-1)$ map respectively A'_{n-1} and A_{n-1} on the right side of o_{n-2}. Finally, $\mathsf{J}'_{n-1}+(0,-2)$ sends $\mathsf{A}_{n-1}+(-2,0) \in \mathsf{O}_{n-2}$ in the lower right zigzag of o_{n-2}.

The case where n is odd can be handled exactly in the same way, except that the octagons are not visited in the same order. Figure 21 summarizes the situation on the straight sides of the octagons, whereas pivot moves on zigzags produce similar intersections than in the even case (see Figure 22).



(a) Case 1

(b) Case 2

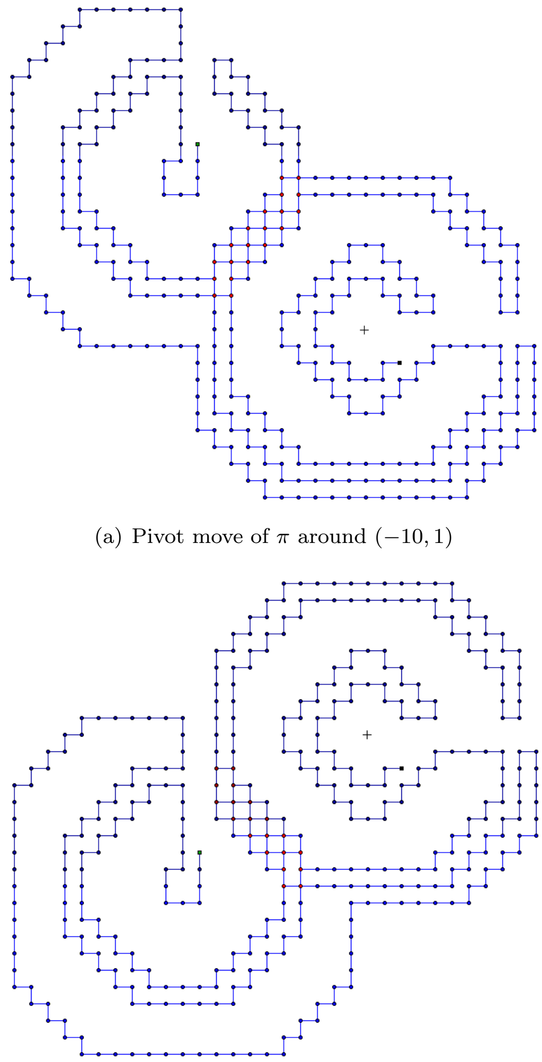
Figure 21: Pivot moves on octagon sides, n odd

6. A list of Open Questions

We enumerate in this section a list of open questions that have appeared interesting to us. Some of them should be very easy to solve, whereas other ones may involve a degree of difficulty.

In the following we define $fSAW^d(n)$ as the class of equivalency of the $n-$step straight walk on $\mathbb{Z}^d$ and $\mathfrak{S}_n^d$ is the equivalent of $\mathfrak{S}_n$ in $\mathbb{Z}^d$. Note that $fSAW^2(n)$ is equal to $fSAW(n)$, as introduced in Definition 3.3.

1. Did these walks constitute an exponentially small subset of SAWs ? For if they do, then the lack of ergodicity in existing software might not be fatal.
2. Is Theorem 3 a consequence of Kesten pattern theorem [22] ?
3. For any dimension d, do we have the existence of $n \in \mathbb{N}^*$ such that $fSAW^d(n) \subsetneq \mathfrak{S}_n^d$?
4. $fSAW^2(2)$ and $fSAW^2(3)$ are obviously connected graphs, but they are not Eulerian. Indeed, more than two vertices have an odd degree both in $fSAW^2(2)$ and $fSAW^2(3)$ (see Figures 14 and 9). Is it the case for all $fSAW^d(n)$?
5. $fSAW^2(2)$ and $fSAW^2(3)$ are Hamiltonian graphs, with the following Hamiltonian circuits:



(a) Pivot move of π around $(-10, 1)$

(b) Pivot move of π around $(-1, -10)$

Figure 22: Case where n is odd (n=1)

- 00 → 03 → 32 → 23 → 10 → 11 → 22 → 33 → 30 → 21 → 12 → 01 → 00 for $fSAW^2(2)$ (see Figure 14).
- 000 → 003 → 010 → 011 → 012 → 001 → 030 → 323 → 330 → 301 → 300 → 333 → 322 → 321 → 332 → 303 → 232 → 233 → 230 → 223 → 212 → 211 → 210 → 221 → 222 → 111 → 110 → 121 → 122 → 123 → 112 → 101 → 100 → 103 → 032 → 033 →

000 for $fSAW^2(3)$ (see Figure 9). In particular, it is possible to find a succession of pivot moves in such a way that, starting from the straight line, the peptide of 4 amino acids visits all the possible conformations exactly once.

Is it a coincidence, or is it the case for every $fSAW^d(n)$?

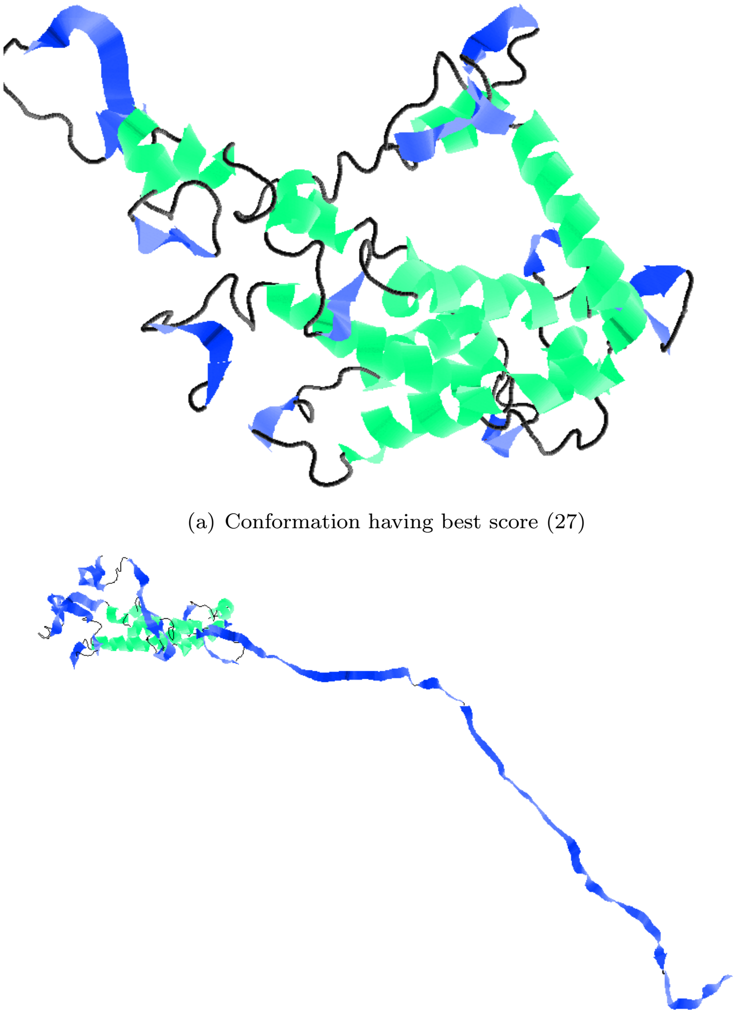
6. What is the exact value of the diameter $D(fSAW^d(n))$?
7. Do we have a connective constant for $fSAW^d(n)$. That is, does the limit $\lim_{n \to +\infty} \phi_n^{1/n}$ exist, and can we bound it ?
8. $u_n = \sharp USAW^d(n)$ is an increasing sequence (for $d = 2$, or for any d)? Does it grow at a given (linear or exponential) rate?
9. Let $k \in \mathbb{N}$. Is the sequence $v_n = \sharp fSAW(n,k)$ increasing with n ? If so at which rate and does it depend on the dimension d? And what about the sequence $w_k = \sharp fSAW(n,k)$ for a given n ?
10. More simply, is there an non-unfoldable walk in $\mathbb{Z}^3$? If so, then PSP software working in 3 dimensions and iterating pivot moves on the straight line cannot reach such a conformation. This is problematic if this conformation is biologically acceptable (or, more dramatically, if it is a conformation that minimizes the free enthalpy). If not, then the results surveyed in this paper for 2D lattices cannot be extended to the general case (3D, without lattice constraints), and the interest of our study, besides being interesting theoretical speaking, is then limited in the context of protein structure prediction.
11. Are the connected components of $\mathfrak{G}_n^d$ convex ? In other words, given two SAWs in a same component C. Are all (or at least one) the shortest paths connecting them on $\mathbb{Z}^d$ in C?
12. Is there a generating function expressing the unfoldable self-avoiding walks more simply, making it possible to enumerate them on the square lattice (like what has been realized in [9]).
13. When we can fold out a self-avoiding walk until a straight line, is it possible to unfold it in such a way that the number of cranks always decreases ? And for two given self-avoiding walks w_i and w_j of the same connected component of $\mathfrak{G}_n$, such that w_i has more cranks than w_j, is there a path from w_i to w_j whose vertices' number of cranks is decreasing ? Is there a relation between the vertex depth and the number of cranks in $\mathbb{Z}^d$?

7. Consequences on Protein Folding

This first theoretical study about (non-)unfoldable self-avoiding walks raises several questions regarding the protein structure prediction problem and the current ways to solve it. In one category of PSP software, the protein is supposed to be synthesized first as a straight line of amino acids. Then this line of a.a. is folded until reaching a conformation that optimizes a given scoring function. By doing so the obtained backbone structures all belong into $fSAW(n)$, where n is the number of residues of the protein. The second category of PSP software consider that, as the protein is already in the aqueous solvent, it does not wait the end of the synthesis to take its 3D conformation. So they consider SAWs whose number of steps increases from 1 to the number of amino acids of the targeted protein and, at each step k, the current walk is stretched (one amino acid is added to the protein) in such a way that the pivot k is placed in the position that optimizes the scoring function they consider. By doing so, the possible predicted backbones are the whole $\mathfrak{G}_3$. The two sets of possible conformations are different, at least when considering 2D low resolution models.

We show by this work that (1) to take place in the first situation (folding the straight line by a succession of pivot moves) can be interesting as the number of possible SAW conformations is smaller than $\sharp\mathfrak{G}_n$. Indeed this interest is directly related to the rate $\dfrac{\sharp fSAW(n)}{\sharp\mathfrak{G}_n} < 1$. If this rate decreases dramatically when n increases, then the computational advantage is obvious. However, we have currently no idea of such a gain, that is, of the growing rate of $\sharp fSAW(n)$ compared to $\sharp\mathfrak{G}_n < 1$. (2) The use of heuristics instead of exact methods (like SAT solvers for instance) is *a priori* not justified for PSP software that fold the straight line. Indeed, the PSP problem has been proven NP hard on the set $\mathfrak{G}_n$ of all possible SAWs. As they consider a strict subset of it, the complexity of the problem might be reduced due to a lower number of cases to consider. Proposition 3 however tends to indicate that this problem still remains difficult in $fSAW(n)$, which nevertheless necessitates a rigorous complexity proof. (3) Biologically speaking, to suppose that the proteins wait to be completely synthesized before starting to fold seems unrealistic, as the synthesis occurs in an aqueous solvent. The protein indeed starts to fold during its synthesis. Furthermore, to the authors' opinion, it is restrictive to consider that the head of the protein definitively stops to fold after having synthesized. Such a supposition is equivalent to make a confusion between local (the SAW at step k) and global (the final optimal SAW) optimization. Indeed, authors of this manuscript recognize honestly that they have no idea to determine if this third approach (continuously folding the walk while stretching it) is more reasonable than the previous ones, and if it is equivalent to either $fSAW(n)$ or to $\mathfrak{G}_n$ (or if it constitutes a third different subset of SAWs). The study of the connected componants however is related to the reachable conformations of already synthesized proteins. Note that non-unfoldable proteins have their conformations limited to the domain of their connected componant. This justify the interest for the study of such walks.

The authors' goal is only to point out the importance to determine the best dynamical system to model protein folding before programming it in PSP software, as this model determine which conformations can be predicted. A last remark to emphasize the importance of such a study: authors of [4] have proven that the dynamical system used in the "folding the straight line" category is chaotic according to Devaney, meaning that any wrong choice of pivot move (due to approximations in the scoring function, for instance) can potentially become dramatic. Other researches ([8] for instance) tend to show that the protein folding process intrinsically embeds a certain amount of chaos. Thus, to



(a) Conformation having best score (27)

(b) Second best conformation (score 24)

Figure 23: Illustration of chaos in protein folding (conformations have been predicted using RaptorX)

use a more or less erroneous model to predict the conformation could have grave consequences in prediction quality. Figure 23 shows the two best conformations predicted by RaptorX [24], a well-known PSP software. We can see that using twice a same model, but with different parameters can potentially lead to quite different conformations, illustrating a possible effect of some chaotic properties exhibited by the chosen model. We can reasonably wonder what is the effect of a wrong model on such a prediction.

8. Conclusion

In this paper, the problem of self-avoiding walks folding in the square lattice has been tackled. Regarding the protein structure prediction problem, we have shown that the set of generated self-avoiding walks depends on the PSP software category. In particular some conformations cannot be reached by just folding the straight line whereas they can be generated using random SAW generators as the pivot algorithm. Starting from this fact, we have proposed a further exploration of the unfoldable self-avoiding walks. Different subsets of self-avoiding walks have been defined, like the set of non-unfoldable walks. We have shown that, even though there is an infinite number of non-unfoldable SAWs, the number of unfoldable SAWs is still exponential. After having described the first obtained results on (non-)unfoldable SAWs, we have proposed a list of open questions that could be explored on these SAWs. Lastly, the link between unfoldable SAWs and proteins has been questioned, and the consequences of the PSP software choice on protein conformation has been highlighted.

Several research problems are interesting to further study and better understand the properties of (non-)unfoldable SAWs, as shown in the open questions section. Our future work will concentrate on studying the connected components of non-unfoldable SAWs as these components define feasable proteins with limited reachable conformations and that cannot straighten. Other interesting questions will be tackled as finding the smallest non-unfoldable SAWs, finding the smallest connected components of non-unfoldable SAWs, and on the optimization of energy levels of a given folded SAW.

Acknowledgments

The authors wish to thank Kamel Mazouzi, Thibaut Cholley, Raphaël Couturier, and Alain Giorgetti for their help in understanding folded and unfoldable SAWs. All the computations presented in the paper have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

References

[1] *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*. IEEE, 2010.

[2] Axel Bacher and Mireille Bousquet-Mélou. Weakly directed self-avoiding walks. *J. Comb. Theory Ser. A*, 118(8):2365–2391, November 2011.

[3] R. Backofen, S. Will, and P. Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding, 1999.

[4] Jacques Bahi, Nathalie Côté, and Christophe Guyeux. Chaos of protein folding. In *IJCNN 2011, Int. Joint Conf. on Neural Networks*, pages 1948–1954, San Jose, California, United States, July 2011.

[5] Jacques Bahi, Nathalie Côté, Christophe Guyeux, and Michel Salomon. Protein folding in the 2D hydrophobic-hydrophilic (HP) square lattice model is chaotic. *Cognitive Computation*, 4(1):98–114, 2012.

[6] Jacques Bahi, Christophe Guyeux, Kamel Mazouzi, and Laurent Philippe. Computational investigations of folded self-avoiding walks related to protein folding. *Journal of Bioinformatics and Computational Biology*, 47(*):246–256, December 2013.

[7] Nicholas R. Beaton, Philippe Flajolet, Timothy M. Garoni, and Anthony J. Guttmann. Some new self-avoiding walk and polygon models. *Fundam. Inf.*, 117(1-4):19–33, January 2012.

[8] Michael Braxenthaler, R. Ron Unger, Ditza Auerbach, and John Moult. Chaos in protein dynamics. *Proteins-structure Function and Bioinformatics*, 29:417–425, 1997.

[9] A. R. Conway, I. G. Enting, and A. J. Guttmann. Algebraic techniques for enumerating self-avoiding walks on the square lattice. *Journal of Physics A Mathematical General*, 26:1519–1534, April 1993.

[10] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding (extended abstract). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 597–603, New York, NY, USA, 1998. ACM.

[11] P. G. de Gennes. Exponents for the excluded volume problem as derived by the Wilson method. *Physics Letters A*, 38(5):339–340, February 1972.

[12] KA Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–9–, March 1985.

[13] Paul J. Flory. The Configuration of Real Polymer Chains. *The Journal of Chemical Physics*, 17(3):303–310, 1949.

[14] Christophe Guyeux, Nathalie M.-L. Côté, Wojciech Bienia, and Jacques Bahi. Is protein folding problem really a NP-complete one? first investigations. *Journal of Bioinformatics and Computational Biology*, 12(1):1350017 (14 pages), February 2014.

[15] Trent Higgs, Bela Stantic, Tamjidul Hoque, and Abdul Sattar. Genetic algorithm feature-based resampling for protein structure prediction. In *IEEE Congress on Evolutionary Computation* [1], pages 1–8.

[16] Md. Hoque, Madhu Chetty, and Abdul Sattar. Genetic algorithm in ab initio protein structure prediction using low resolution model: A review. In Amandeep Sidhu and Tharam Dillon, editors, *Biomedical Data and Applications*, volume 224 of *Studies in Computational Intelligence*, pages 317–342. Springer Berlin Heidelberg, 2009.

[17] Dragos Horvath and Camelia Chira. Simplified chain folding models as metaheuristic benchmark for tuning real protein folding algorithms? In *IEEE Congress on Evolutionary Computation* [1], pages 1–8.

[18] Barry D. Hughes. *Random walks and random environments, Volume 1: Random walks.* Clarendon Press, Oxford, March 1995.

[19] Md. Kamrul Islam and Madhu Chetty. Clustered memetic algorithm for protein structure prediction. In *IEEE Congress on Evolutionary Computation* [1], pages 1–8.

[20] Iwan Jensen. Enumeration of self-avoiding walks on the square lattice. *J. Phys. A*, pages 5503–5524, 2004.

[21] Iwan Jensen. Improved lower bounds on the connective constants for two-dimensional self-avoiding walks. *Journal of Physics A: Mathematical and General*, 37(48):11521+, 2004.

[22] Neal Madras and Alan D. Sokal. The pivot algorithm: A highly efficient monte carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50:109–186, 1988.

[23] Neal Noah Madras and Gordon Slade. *The self-avoiding walk.* Probability and its applications. Birkhäuser, Boston, 1993.

[24] Jian Peng and Jinbo Xu. Raptorx: Exploiting structure information for protein alignment by statistical inference. *Proteins*, 79(S10):161–171, 2011.

[25] Alena Shmygelska and Holger Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6(1):30, 2005.

[26] Gordon Slade. The self-avoiding walk: a brief survey. Blath, Jochen (ed.) et al., Surveys in stochastic processes. Selected papers based on the presentations at the 33rd conference on stochastic processes and their applications, Berlin, Germany, July 27–31, 2009. Zürich: European Mathematical Society (EMS). EMS Series of Congress Reports, 181-199 (2011)., 2011.

[27] Ron Unger and John Moult. Genetic algorithm for 3d protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.