The non unfoldable self-avoiding walks

Christophe Guyeux

FEMTO-ST - DISC Department - AND Team

March 21th, 2014

Plan

The PSP problem

Introducing the foldable SAWs

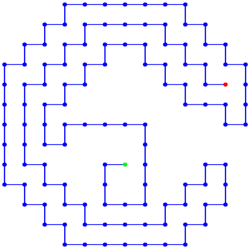
The study of foldable SAWs

Conclusion

Self-Avoiding Walk

Let $d \geqslant 1$. A $n-$step *self-avoiding walk* (SAW) 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 $w(n) = y$,
- $|w(i+1) - w(i)| = 1$,
- $\forall i, j \in [\![0, n]\!], i \neq j \Rightarrow w(i) \neq w(j)$ (self-avoiding property).



Protein Structure Prediction problem

The Protein Folding Process

- Proteins, polymers formed by different kinds of amino acids, fold to form a specific tridimensional shape
- This geometric pattern defines the majority of functionality within an organism
- Contrary to the mapping from DNA to the amino acids sequence, the complex folding of this last sequence still remains not well-understood

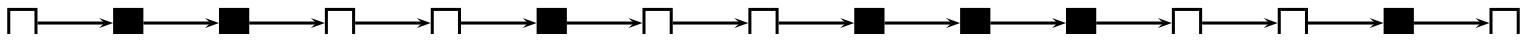
The 2D HP model

Hydrophilic-hydrophobic 2D square lattice model:

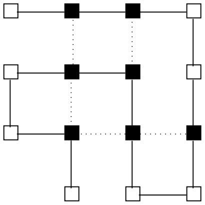
- A protein conformation is a “self-avoiding walk (SAW)” on a 2D lattice (low resolution model)
- Its free energy *E* must be minimal
- Hydrophobic interactions dominate protein folding:
  - Protein core freeing up energy is formed by hydrophobic amino acids
  - Hydrophilic a.a. tend to move in the outer surface
- *E* depends on contacts between hydrophobic amino acids that are not contiguous in the primary structure

The 2D HP model

Objective: to map the labeled straight line



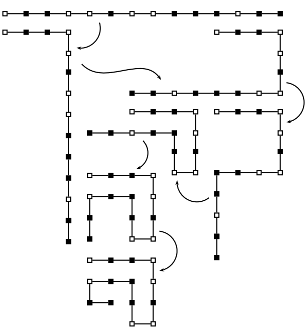
in this latter, having more black neighbors:

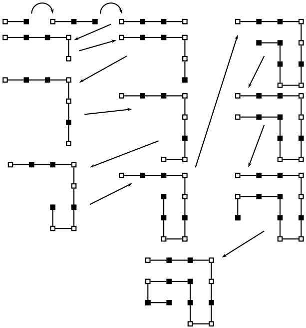


Resolving the PSP problem

- Being NP-complete, the optimal conformation(s) cannot be found exactly for large *n*'s
- Conformations are thus *predicted* using AI tools
- Some strategies found in the literature:
    1. start by predicting the 2D backbone,
    2. then refine the obtained conformation in a 3D shape
- At least two strategies for 2D backbone prediction:
    - Method 1: iterating $\pm 90^\circ$ pivot moves on the straight line
    - Method 2: stretching 1 amino acid until obtaining an *n*-steps conformation
    - ...?

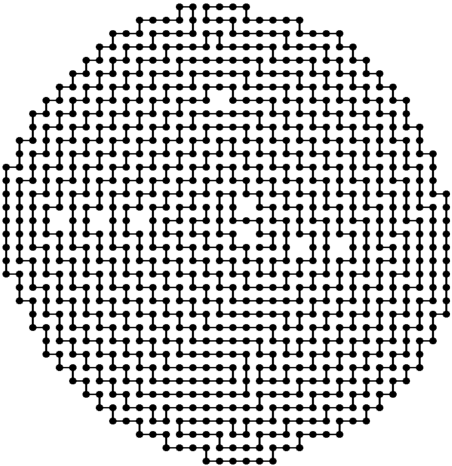
Various methods for solving PSP





1. PSP by folding SAWs
2. PSP by stretching SAWs

My first example

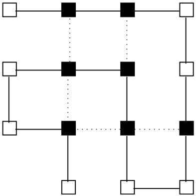


Introducing the foldable SAWs

Self-avoiding walk encoding

Absolute encoding of a SAW:

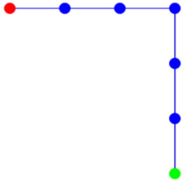
Movement	Encoding
Forward $\rightarrow$	0
Down $\downarrow$	1
Backward $\leftarrow$	2
Up $\uparrow$	3



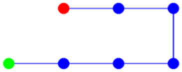
Absolute encoding:
00011123322101

Pivot move of $\pm 90^\circ$

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



(a) 000111



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

A $\pm 90^\circ$ pivot move applies this function on the tail of the walk

Madras and Sokal

Theorem

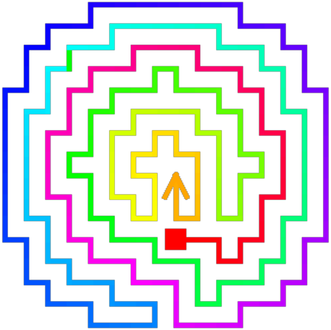
The pivot algorithm is ergodic for self-avoiding walks on $\mathbb{Z}^d$ provided that all axis reflections, and:

- either all 90° rotations
- or all diagonal reflections,

are given nonzero probability.

Any $N-$step SAW can be transformed into a straight rod by some sequence of $2N - 1$ or fewer such pivots.

Madras and Sokal example



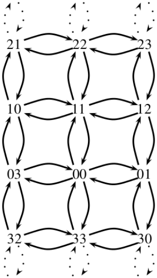
Ergodicity is lost when considering single $\pm 90^\circ$ pivot moves

A graph structure for unfolded SAWs

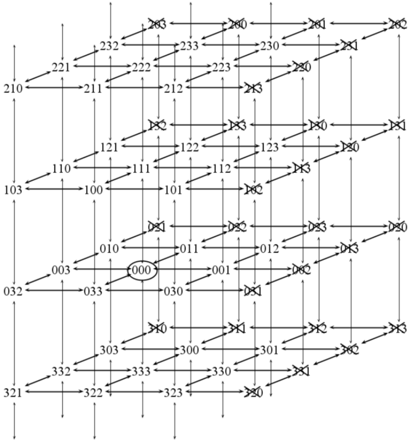
The graph $\mathfrak{G}_n$ is defined as follows:

- its vertices are the $n-$step self-avoiding walks, described in absolute encoding;
- there is an edge between two vertices s_i, s_j $\Leftrightarrow$ s_j can be obtained by one pivot move of $\pm 90^\circ$ on s_i.

Examples of $\mathfrak{G}_n$



$\mathfrak{G}_2$



$\mathfrak{G}_3$

Method 1 *vs* method 2

- $\mathcal{S}_n$: all the vertices of $\mathfrak{G}_n$ (all n-step SAWs)
  $\Rightarrow$ An equivalence relation: $w_1 \mathcal{R}_n w_2 \Leftrightarrow w_1$ is in the same connected component that w_2 on $\mathfrak{G}_n$.
- $fSAW_n$: the connected component of the straight line $00\ldots0$ in $\mathfrak{G}_n$,

Method 1 *vs* method 2

- $\mathcal{S}_n$: all the vertices of $\mathfrak{G}_n$ (all n-step SAWs)
  $\Rightarrow$ An equivalence relation: $w_1 \mathcal{R}_n w_2 \Leftrightarrow w_1$ is in the same connected component that w_2 on $\mathfrak{G}_n$.
- $fSAW_n$: the connected component of the straight line $00\ldots0$ in $\mathfrak{G}_n$,

We rediscovered that for some n, $fSAW_n \subsetneq \mathfrak{G}_n$.

- It is an obvious consequence of Madras example
- This fact is not known by some computer scientists
- $\Rightarrow$ Method 1 and Method 2 do not produce the same set of conformations

Method 1 *vs* method 2

- $\mathcal{S}_n$: all the vertices of $\mathfrak{G}_n$ (all n-step SAWs)
  $\Rightarrow$ An equivalence relation: $w_1 \mathcal{R}_n w_2 \Leftrightarrow w_1$ is in the same connected component that w_2 on $\mathfrak{G}_n$.
- $fSAW_n$: the connected component of the straight line $00\ldots0$ in $\mathfrak{G}_n$,

We rediscovered that for some n, $fSAW_n \subsetneq \mathfrak{G}_n$.

- It is an obvious consequence of Madras example
- This fact is not known by some computer scientists
- $\Rightarrow$ Method 1 and Method 2 do not produce the same set of conformations

How evolves the ratio $\dfrac{\sharp fSAW_n}{\sharp \mathfrak{G}_n}$?

Some subsets of SAWs

We introduce the following sets:

- $fSAW_n$ is the equivalence class of the $n-$step straight walk, or the set of all folded 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 unfoldable walks.
  $\Rightarrow$ Madras' walk belongs in $USAW_{223}$
- 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 $\pm 90°$.

The study of foldable SAWs

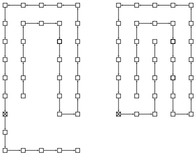
Current investigation techniques

- For small *n*'s: brute force.
  - Nb of $fSAW(n)$ = 4*Nb of $fSAW(n)$ starting by 0 = 4*(Nb of $fSAW(n)$ starting by 00 + 2* Nb of $fSAW(n)$ starting by 01)
  - Stop when a polyomino appears
- For large *n*'s: backtracking on reduced human solutions

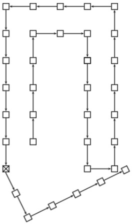
A short list of results

1. $2^{n+2} \leqslant \sharp fSAW_n \leqslant 4 \times 3^n$
2. $\forall n \leqslant 22, fSAW_n = \mathfrak{G}_n$ ($n \leqslant 11$ in triangular lattice)
3. $fSAW_{108} \subsetneq \mathfrak{G}_{108}$.
   - let ν_n the smallest $n \geqslant 2$ such that $USAW_n \neq \emptyset$. Then $23 \leqslant \nu_n \leqslant 108$.
   - We can obtain all $\mathfrak{G}_n, n \leqslant 22$ by increasing the number of cranks
4. $\forall n \leqslant 28, f^1SAW_n = \mathfrak{G}_n$, while $f^1SAW_{108} \subsetneq \mathfrak{G}_{108}$.
5. $\exists k > 2$ such that $fSAW(n,k)$ is nonempty.
6. The diameter of $fSAW(n)$ is equal to $2n$.

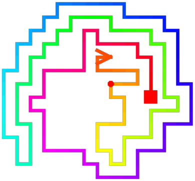
$fSAW_n$ is not $fSAW'_n$



Acceptable in *fSAW*

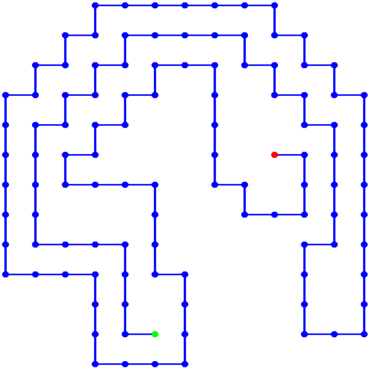


Not in *fSAW′*

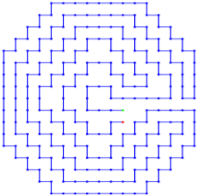


$fSAW_n \neq fSAW'_n$

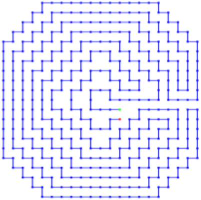
Current smallest (108-step) USAW



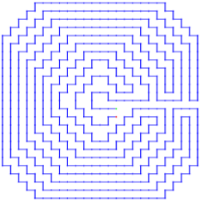
$\sharp\{n \mid USAW(n) \neq \emptyset\} = \infty$



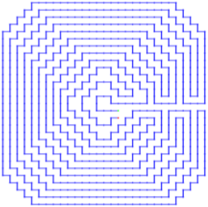
(a) s_0 (239-step walk)



(b) s_1 (391-step walk)



(c) s_2 (575-step walk)



(d) s_3 (791-step walk)

Cardinalities of subsets of SAWs

n	$\sharp\mathfrak{S}_n$	$\sharp f^1SAW(n)$	$\sharp USAW(n) = \sharp\overline{f^1SAW(n)}$	$\sharp fSAW(n)$
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	6416596
16	17245332	17245332	0	17245332
17	46466676	46466676	0	46466676
18	124658732	124658732	0	124658732
19	335116620	335116620	0	335116620
20	897697164	897697164	0	897697164
21	2408806028	2408806028	0	2408806028
22	6444560484	6444560484	0	6444560484
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	?	?	?

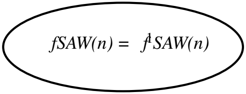
Cardinalities of subsets of SAWs

107	?	?	$\geqslant 3$	?
108	?	?	$\geqslant 1$	?
111	?	?	$\geqslant 5$	?
112	?	?	$\geqslant 1$	?
113	?	?	$\geqslant 2$	?
114	?	?	$\geqslant 2$	?
115	?	?	$\geqslant 5$	?
116	?	?	$\geqslant 3$	?
117	?	?	$\geqslant 4$	?
118	?	?	$\geqslant 2$	?
119	?	?	$\geqslant 2$	?
121	?	?	$\geqslant 4$	?
122	?	?	$\geqslant 5$	?
123	?	?	$\geqslant 1$	?
132	?	?	$\geqslant 7$	?
133	?	?	$\geqslant 6$	?
134	?	?	$\geqslant 95$	?
135	?	?	$\geqslant 165$	?
136	?	?	$\geqslant 40$	?
137	?	?	$\geqslant 50$	?
138	?	?	$\geqslant 175$	?
139	?	?	$\geqslant 179$	?
140	?	?	$\geqslant 66$	?
141	?	?	$\geqslant 119$	?
142	?	?	$\geqslant 322$	?
143	?	?	$\geqslant 476$	?
144	?	?	$\geqslant 8$	?
145	?	?	$\geqslant 18$	?
146	?	?	$\geqslant 54$	?
235	?	?	$\geqslant 1$	?
239	?	?	$\geqslant 1$	?
391	?	?	$\geqslant 1$	?
575	?	?	$\geqslant 1$	?

Case of triangular SAWs

n	saw(n)	$\sharp f^1 SAW(n)$
0	1	1
1	6	6
2	30	30
3	138	138
4	618	618
5	2730	2730
6	11946	11946
7	51882	51882
8	224130	224130
9	964134	964134
10	4133166	4133166
11	17668938	17668938
12	75355206	
13	320734686	
14	1362791250	
15	5781765582	
16	24497330322	
17	103673967882	
18	438296739594	

Vien diagrams for some $\mathfrak{G}_n$



$\mathfrak{G}_n$ for $n \leqslant 22$

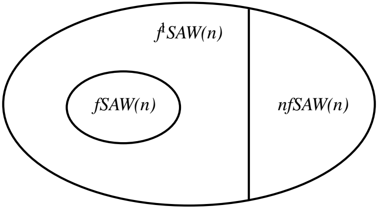
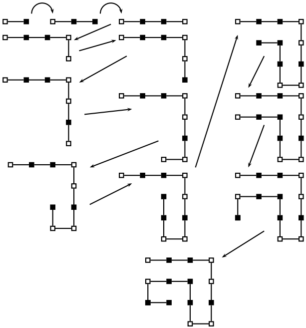


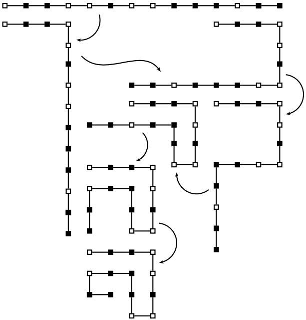
Diagram of $\mathfrak{G}_n$ for $n = 108$

Conclusion

All walks are interesting ?



Protein synthesis



Intrinsically complicated prot.

Some open questions

1. Did these walks constitute an exponentially small subset of SAWs ?
2. The PSP problem still remains NP-complete in $fSAW_n$?
3. For any dimension d, do we have the existence of $n \in \mathbb{N}^*$ such that $fSAW_n^d \subsetneq \mathfrak{G}_n^d$?
4. $fSAW_2^2$ and $fSAW_3^2$ are Hamiltonian graphs, but they are not Eulerian. What about $fSAW_n^k$?
5. is there an unfoldable walk in $\mathbb{Z}^3$?
6. Are the connected components of $\mathfrak{G}_n^d$ convex ?
7. ...

Other open questions

- Monte-Carlo approach ?
- Genetic algorithm approach ?
- Dynamic programming ?
- Pivot algorithm ?
- Forbidden patterns ?

Thank you!

Any question/suggestion/idea ?

christophe.guyeux@univ-fcomte.fr