Università della Svizzera italiana

Faculty of Informatics

USI Technical Report Series in Informatics

An effective exact algorithm and a new upper bound for the number of contacts in the hydrophobic-polar 2D-lattice model

Emanuele Giaquinta[1], Laura Pozzi[2]

[1] Department of Computer Science, University of Helsinki, Finland
[2] Faculty of Informatics, Università della Svizzera italiana, Switzerland

Abstract

Protein Structure Prediction (PSP) is the problem of predicting the three-dimensional native structure of a protein given its primary structure, i.e., the corresponding sequence of amino acids. Different approaches have been proposed to *model* this problem, and this research explores the prediction of optimal structures using the well studied simplified lattice Hydrophobic and Polar (HP) model—in particular, in the 2D square lattice.
We present a twofold result. First, we devise a new upper bound for the number of contacts achievable by an HP sequence, and show that it is in several cases more stringent than the upper bound previously known in literature. Then, we present an innovative algorithm that outperforms the state of the art in *exact* approaches for the prediction of optimal structures in lattice protein model, for *2-D* square lattices. The algorithm, called MINWALK and based on a heavily *pruned* exhaustive search, also outperforms the state of the art in *non-exact* approaches in several cases.
Due to this algorithm, it is now possible to *prove optimal* results in the square 2D lattice, for standard HP sequences of size up to 80 elements, which were only best-known-results previously. Furthermore, we provide the degeneracy (i.e. all optimal solutions) of such benchmark sequences, which was unknown in literature. These results can be a useful tool to foster advances in further research.

Report Info

Published
November 2012

Number
USI-INF-TR-2012-2

Institution
Faculty of Informatics
Università della Svizzera italiana
Lugano, Switzerland

Online Access
www.inf.usi.ch/techreports

1 Introduction

Protein Structure Prediction (PSP) is the problem of predicting the three-dimensional native structure of a protein given its primary structure, i.e., the corresponding sequence of amino acids. It is among the most challenging problems in computational biology and, despite many advancements, it is still an open challenge. In the denatured state a protein is a linear chain of amino acids. The physical process by which a protein achieves its final state, also known as tertiary structure, is called protein folding. In particular, this state corresponds to the folded state with minimum free energy.

Different approaches have been proposed to *model* this problem. This research aims at exploring the prediction of optimal structures using the simplified lattice model. In this model, each amino acid is represented as a point in an integer lattice and the PSP problem is reduced to that of finding the *self-avoiding walk*(s) of minimum energy on the lattice. The energy model used here is the HP model, due to Lau and Dill [8], where one considers the *hydrophobic effect* as the main interaction in the folding process. Because of this effect, the hydrophobic amino acids present in the folded proteins tend to form a compact core in the inside, while the polar amino acids tend to group on the surface. In this model each amino acid is encoded as a bead, which can be of two different types: **Hydrophobic** (H) and **Polar** (or hydrophilic) (P). Two beads of type H that are

topologically first-neighbours and are not connected in the sequence contribute negatively to the energy. All the other combinations contribute zero energy. The PSP problem on 2-D and 3D lattices is NP complete [4, 3].

In this paper we present new properties of HP sequences in *2-D* square lattices, and an innovative algorithm that outperforms the state of the art in *exact* approaches for *2-D* square lattices. In particular, we describe a new relationship between the energy and the area of a specific type of configurations that we call *connected* and, on top of this result, *a new upper bound* on the energy achievable by an HP sequence. The algorithm, called MINWALK, is based on a *pruned* exhaustive search of the input space. Given an HP sequence, our algorithm returns all the optimal configurations of the sequence.

Concerning a comparison with the state of the art in the field, an interesting survey of combinatorial algorithms on lattice protein folding models is [7]. A useful characterisation is to divide past efforts into the following categories: *exact* and *non-exact* approaches on 2-D and/or 3-D lattices. In the remainder of this work, whenever we mention 2-D or 3-D, we mean 2-D *square* lattices and 3-D *cubic* lattices, unless stated otherwise.

For *exact* approaches, the state of the art in 2-D is represented by [6] (there have been other, earlier approaches [13], which, however, can only deal with smaller instances of the problem). An exact algorithm [6] was proposed that scales up to a an instance size equal to 25. This algorithm enumerates, for a given length, *all* non-degenerate sequences (i.e., the ones that admit a *single* optimal configuration) of that length, together with the optimal configuration itself. In contrast, the algorithm presented in this paper finds all optimal (possibly degenerate) configurations when given an input HP sequence, but it scales for rather larger sizes: it can find optimal solutions to random sequences of length up to around 45, and to specific sequences of length up to 85.

Several non-exact approaches also exist, to find *good* structures of a given HP sequence [16, 10, 9, 5, 14, 15, 12]. The papers [5, 14, 15, 12] illustrate, to the best of our knowledge, the best-performing *non-exact* approaches in the literature. Ref. [15] contains an extensive comparison of them. In this paper, we compare our exact algorithm with the best performing non-exact approaches, and we show that, for most benchmarks of sizes up to 85, we exhibit similar or even better running times, while *guaranteeing* the optimality of the solutions found.

To sum up: we raise the maximum size of sequences that can be dealt with by an exact algorithm to return all its optimal configurations in 2-D. At present our algorithm returns optimal structures for sequences up to size around 45 in hundreds of seconds. We also present running times for several benchmark sequences of size 30, mostly in fractions of seconds. There is no exact approach for the 2-D square lattice that can return optimal configurations for these sizes. We also show by experimental evaluation that our algorithm is competitive if compared with the best non exact methods [5, 14, 15, 12]. In all but one case, it achieves running times comparable with, or better than, the ones obtained by these algorithms, with the additional advantage that it guarantees optimality.

Several studies in structural biology look at the *designability* of a structure, i.e. the number of different sequences that have that structure as its unique minimum, both at simple exact models level, in the 2-D lattice [6] and in the 3-D lattice [11], and also at the atomic level. Only exact approaches can be used to study degeneracy. In fact, since non-exact approaches cannot guarantee the optimality of the solution given, they cannot obviously, as a consequence, give the degeneracy of optimal solutions. The algorithm here proposed can be a useful tool to give new insights into this issue in the 2-D lattice.

For the 3-D lattice case, the exact method proposed in [2] represents the state of the art. The authors introduced a high performance algorithm, based on constraint programming, that can find optimal structures for sequences of length higher than 100. While the algorithm works on both the 3-D cubic and FCC lattices, it does not extend to 2-D. Most likely, this is because it depends on enumeration of all possible cores, and it is therefore particularly suited to lattices where optimal configurations result in extremely compact core. In the 2-D lattice, this is not always true. The algorithm proposed here can in principle also be extended to the 3-D lattice, but this has not been done yet and therefore a comparison with [2] is not feasible.

The paper is organized as follows. In Section 2 we introduce the notations and terminology used in the paper. In Section 3 we discuss new results on HP sequences in 2-D lattice. In Section 4 we present a new algorithm for the protein folding problem in the HP model in 2-D lattice. In Section 5 we present experimental results to assess the performance of our algorithm and compare it with the existing methods. Finally, we draw our conclusions in Section 6.

2 Preliminary definitions

Let Σ be a finite alphabet of symbols and Σ^* the Kleene star of Σ, i.e., the set of all possible finite sequences of symbols belonging to Σ. An amino acid sequence of length n is a sequence $a_1,\ldots,a_n \in (\{A,\ldots,Z\} \setminus \{B,J,O,U,X,Z\})^*$. The HP encoding of an amino acid sequence $a_1,\ldots,a_n$ is the sequence $\hat{a}_1,\ldots,\hat{a}_n \in \{H,P\}^*$, where

$$\hat{a}_i = \begin{cases} H & \text{if } a_i \in \{A,V,L,I,P,F,M,W,G,C\} \\ P & \text{otherwise} \end{cases},$$

for $i = 1,\ldots,n$. We encode a self-avoiding walk on a 2-D lattice of an HP sequence $\hat{a}_1,\ldots,\hat{a}_n$ as the sequence $\langle d_1,\ldots,d_{n-1}\rangle$, where $d_i \in \{L,R,U,D\}$ is the direction of the link that connects beads $\hat{a}_i$ and $\hat{a}_{i+1}$. Observe that, for $i \geq 2$, each d_i can be chosen in at most three ways because there are at most three empty cells that are first-neighbours to bead $\hat{a}_{i-1}$. Moreover, the second bead, and its associated direction d_1, can be fixed at any of the four first-neighbour sites that are all equivalent by symmetry. Given a configuration $\langle d_1,\ldots,d_{n-1}\rangle$, we define the vector (x_i,y_i) as the coordinate of bead $\hat{a}_i$ in the lattice, where

$$(x_i,y_i) = \begin{cases} (0,0) & \text{if } i = 1 \\ (x_{i-1}-1, y_{i-1}) & \text{if } i > 1 \text{ and } d_{i-1} = L \\ (x_{i-1}+1, y_{i-1}) & \text{if } i > 1 \text{ and } d_{i-1} = R \\ (x_{i-1}, y_{i-1}-1) & \text{if } i > 1 \text{ and } d_{i-1} = U \\ (x_{i-1}, y_{i-1}+1) & \text{if } i > 1 \text{ and } d_{i-1} = D \end{cases}$$

Given two distinct beads $\hat{a}_i$ and $\hat{a}_j$, $i \neq j$, we define the energy of the corresponding contact, if any, as $\epsilon(i,j) = C(i,j)E(\hat{a}_i,\hat{a}_j)$, where

$$C(i,j) = \begin{cases} 1 & \text{if } |i-j| \neq 1 \text{ and } d((x_i,y_i),(x_j,y_j)) = 1 \\ 0 & \text{otherwise} \end{cases}$$

indicates whether the beads are topologically first-neighbours ($d(\cdot)$ is the euclidean distance) and are not adjacent in the original sequence and

$$E(\hat{a}_1,\hat{a}_2) = \begin{cases} -1 & \text{if } \hat{a}_1 = \hat{a}_2 = H \\ 0 & \text{otherwise} \end{cases}$$

defines the energy of a contact depending on the bead types. In the HP model an HH contact provides a unitary contribution while all the other combinations yield zero energy. Finally, we define the energy of a configuration as

$$\sum_{i<j} \epsilon(i,j)$$

In this model the problem of finding the configuration(s) with minimal energy is equivalent to finding the configuration(s) with maximum number of HH contacts.

Given an HP sequence $S = \hat{a}_1,\ldots,\hat{a}_n$, we partition the set of H beads into the following two subsets:

$$\mathcal{H}_E = \{i \mid \hat{a}_i = H \text{ and } i \bmod 2 = 0\}$$
$$\mathcal{H}_O = \{i \mid \hat{a}_i = H \text{ and } i \bmod 2 = 1\}$$

We say that an H bead is of type E (O) if it belongs to the subset $\mathcal{H}_E$ ($\mathcal{H}_O$). The $\mathcal{H}_E$ set contains all H beads of even index, and the $\mathcal{H}_O$ set contains all H beads of odd index. It is easy to see that on a 2-D lattice an HH contact is possible only between two beads that do not belong to the same subset, as no pair of beads in either subset can be first-neighbours. This property is known as the parity problem and holds also in cubic lattices. Given a bead $\hat{a}_i$, for $i = 1,\ldots,n$, we say that the bead is internal if $i \notin \{1,n\}$. Note that, in any configuration that is a valid self-avoiding walk, all internal beads have exactly two sequence-adjacent elements, while the first and the last bead have exactly one sequence-adjacent element. The following elementary lemma holds:

Lemma 1. *Given an HP sequence of length n, the maximum number of non-adjacent first neighbours of a given bead, in any self-avoiding walk covering the whole sequence, is equal to 2 if the bead is internal and to 3 otherwise, i.e.,*

$$\sum_j C(i,j) \leq \begin{cases} 2 & \textit{if } i \notin \{1,n\} \\ 3 & \textit{otherwise} \end{cases}$$

for all $i = 1, \ldots, n$.

Hence, the maximum number of contacts in which a bead can participate is equal to 2 if the bead is internal and to 3 otherwise. The maximum number of contacts achievable by the beads of type O and E is then

$$\begin{aligned} C_E &= 2|\mathcal{H}_E| + e_E \\ C_O &= 2|\mathcal{H}_O| + e_1 + e_O \end{aligned} \tag{1}$$

where the e_x terms take care to consider extra contacts made by sequence extremities: e_E is 1 if $\hat{a}_n \in \mathcal{H}_E$ and 0 otherwise, e_O is defined analogously to e_E, and e_1 is 1 if $\hat{a}_1 = H$ and 0 otherwise. The maximum number of contacts achievable by the HP sequence S is then bounded with the number

$$C_{\text{parity}} = \min(C_E, C_O) \tag{2}$$

However, in the next section we present *a new upper bound* on the number of contacts achievable by an HP sequence.

3 New results on the structures of HP sequences

In this section we present some new results on the configurations of HP sequences in the 2-D lattice model. Given a configuration of an HP sequence on a 2-D lattice, we define the **H-core** as the smallest rectangle containing all the H beads. Let $S = \hat{a}_1, \ldots, \hat{a}_n$ be an HP sequence with m H beads and let also

$$\mathcal{P}_S = \{i \mid 2 \leq i < n \wedge \hat{a}_i = P \wedge \hat{a}_{i-1} = \hat{a}_{i+1} = H\}$$

be the set of P beads whose previous and next neighbours in the sequence are H beads. We call such beads P singlets. Observe that the minimum area that the H-core must span is equal to

$$A_{\min} = m + |\mathcal{P}_S|$$

because it must contain, by definition, all the H beads, as well as all the P beads in $\mathcal{P}_S$, since it is not possible to place them outside the core.

We now introduce the concept of a *connected* H-core, which in turn will lead to the definition of a new upper bound on the number of contacts achievable by an HP sequence. We say that an H-core is *connected* if it satisfies the property that each row and column contains at least one H bead or one P singlet. We devise an interesting relationship between the size of a *connected* H-core and the maximum number of contacts that can be achieved within it. Let (L_1, L_2) be the dimensions of an H-core. The number of cells available in such a rectangle is $L_1 L_2$ and the number of segments in such a rectangle, which represents the maximum number of contacts that can be obtained by placing an H bead in each cell, is

$$(L_1 - 1)L_2 + (L_2 - 1)L_1 .$$

This can be observed in Figure 1a, depicting an example H-core of dimensions $L_1 = 3$ and $L_2 = 4$, correspondingly containing 12 cells ($3 * 4$), and 17 segments ($2 * 4 + 3 * 3$) linking these cells. These segments represent the potential contacts that a given configuration, within such H-core, can realise.

Observe that in an H-core with dimensions (L_1, L_2), m cells will be occupied by H beads, and therefore $L_1 L_2 - m$ cells must be either empty or occupied with P beads. The segments adjacent to these cells cannot contribute any contact. Figure 1b shows a sequence (HHHPHHHHPHP) configured into the example H-core; and for such configuration, Figure 1c analyses the nature of the available 17 segments. Four (in red) are realised into contacts, while the remaining 13 are not. Of the remaining 13: five (in black) are H-H links internal to the sequence, and eight (in light grey) are adjacent to non-H cells segments. We call these 'wasted'. Figure 1d and 1e show a different configuration for the same sequence, which realises only 1 contact: for the remaining segments, five are H-H links internal to the sequence (this number is of course constant to the sequence and does not depend on the configuration) and as many as 11 segments are wasted.

In order to estimate the *minimum number of segments that must be wasted* (i.e. they are not realised into contacts) because of non-H cells, we introduce the following simple result:

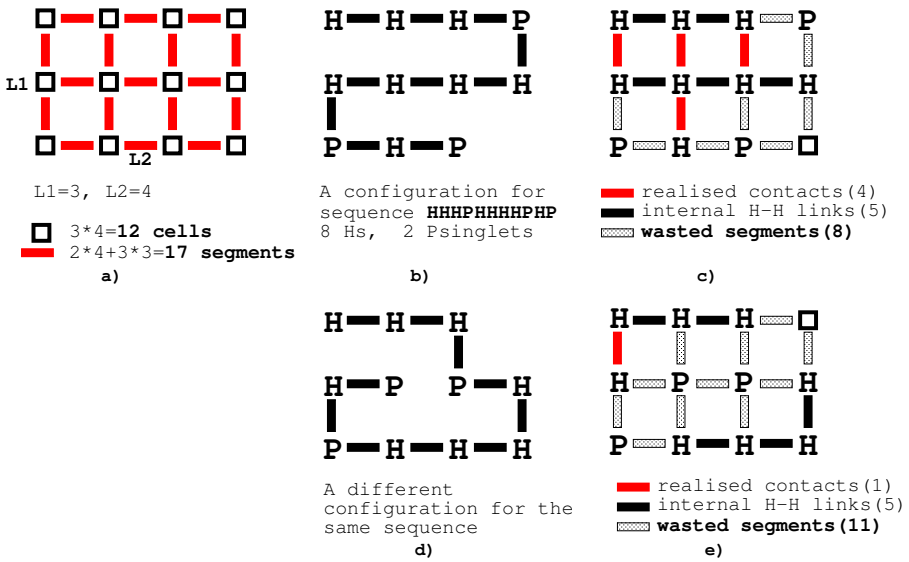

Figure 1: a) Number of cells and of segments for an H-core of dimensions $L_1 = 3$ and $L_2 = 4$. b) A configuration for sequence HHHPHHHHPHP c) Analysis of such configuration, in terms of segments types: 4 segments are realised into contacts, 5 are internal H-H links, 8 segments are wasted). d) A different configuration for the same sequence e) Analysis of such configuration (now only 1 contact is realised, and 11 are wasted.

Lemma 2. *Let $\bar{P}$ denote either a P bead or an empty cell on the lattice. Given an H-core and an associated row (column) with $n_{\bar{P}} > 0$ beads of type $\bar{P}$, the number of H$\bar{P}$ and $\bar{P}\bar{P}$ pairs of adjacent beads in the row (column) is equal to $n_{\bar{P}} - 1 + n_H$, where n_H is the number of H beads after collapsing all the sequences of adjacent H beads into a single H bead.*

Proof. Consider the row (column) obtained by collapsing all the sequences of adjacent H beads into a single H bead; this normalization removes all the HH pairs, making the analysis simpler, and does not change the number of $\bar{P}\bar{P}$ and H$\bar{P}$ pairs. Let h_e be the number of H beads that are adjacent, in the normalized sequence, to one $\bar{P}$ bead, i.e., the number of H beads that are the first or last element of the row (column); $n_H - h_e$ is thus the number of H beads that are adjacent to two $\bar{P}$ beads. Then, it is easy to verify that the number of $\bar{P}\bar{P}$ and H$\bar{P}$ pairs is equal to $n_{\bar{P}} - 1 - (n_H - h_e)$ and $2(n_H - h_e) + h_e$, respectively. Hence, the total number of pairs is $n_{\bar{P}} - 1 + n_H$ and the claim follows. □

Let $n_{\bar{P}}(i,*)$ and $n_{\bar{P}}(*,j)$ denote the number of beads of type $\bar{P}$ in the i-th row and in the j-th column, respectively. Likewise for $n_H(i,*)$ and $n_H(*,j)$. Let also $R_P = \{1 \le i \le L_1 \mid n_{\bar{P}}(i,*) > 0\}$ and $C_P = \{1 \le j \le L_2 \mid n_{\bar{P}}(*,j) > 0\}$ be the set of row and column indexes with at least one $\bar{P}$ bead. Based on Lemma 2 we are now able to provide a *lower bound on the number of wasted contacts in a connected H-core*:

Theorem 1. *Given a connected H-core with dimensions (L_1, L_2) and containing m H beads, the number of H$\bar{P}$ and $\bar{P}\bar{P}$ pairs is at least $2(L_1 L_2 - m)$.*

Proof. By Lemma 2 the total number of H$\bar{P}$ and $\bar{P}\bar{P}$ pairs is equal to

$$\sum_{i \in R_P} n_{\bar{P}}(i,*) + \sum_{j \in C_P} n_{\bar{P}}(*,j) + \sum_{i \in R_P} (n_H(i,*) - 1) + \sum_{j \in C_P} (n_H(*,j) - 1).$$

It is easy to see that the first part of the expression, $\sum_{i \in R_P} n_{\bar{P}}(i,*) + \sum_{j \in C_P} n_{\bar{P}}(*,j)$, is equal to $2(L_1 L_2 - m)$, as each $\bar{P}$ bead occurs once in all the rows and once in all the columns. We now show that, in a connected H-core, the second part, $\sum_{i \in R_P} (n_H(i,*) - 1) + \sum_{j \in C_P} (n_H(*,j) - 1)$, is ≥ 0. Consider a generic row with index i (the reasoning is analogous in the case of a column). If $n_H(i,*) > 0$ then its contribute to the sum is at least 0. Instead, if $n_H(i,*) = 0$, by definition of connected H-core, the row must contain a P singlet. Let j be the column coordinate of the P singlet. Since row i does not contain any H bead, the two H beads adjacent (in the sequence) to the P singlet must belong to column j, i.e., $n_H(*,j) \ge 2$. Hence, the total contribution of row i and column j to the sum is equal to $n_H(i,*) + n_H(*,j) - 2 \ge 0$. □

Therefore, by Theorem 1, in a connected H-core, the number of wasted contacts is at least $2(L_1L_2-m)$. Moreover, we have also one contact less for each H-H link in the walk because these links must lie inside the H-core. Let

$$\mathcal{H}_I = \{i \mid i < n \wedge \hat{a}_i = \hat{a}_{i+1} = H\}$$

be the set of H beads such that the next adjacent bead in the sequence is of type H. The number of H-H links only depends on the sequence and is equal to $|\mathcal{H}_I|$ (this value is 5, for the example in Figure 1). For the sequence S an upper bound of the number of contacts achievable in a connected H-core with dimensions (L_1, L_2) is then

$$(L_1-1)L_2+(L_2-1)L_1-2(L_1L_2-m)-|\mathcal{H}_I| = 2m-|\mathcal{H}_I|-L_1-L_2\,.$$

Let

$$area(t) = \lfloor t/2 \rfloor \lceil t/2 \rceil\,, \tag{3}$$

be the maximum area of a rectangle with semi-perimeter t. By coalescing the two variables L_1 and L_2 into the corresponding sum, denoted as t, we can then define the function

$$\phi(t) = 2m-|\mathcal{H}_I|-t\,,$$

that maps a semi-perimeter onto the maximum number of contacts achievable in an H-core with area at most equal to $area(t)$. The ϕ function is strictly decreasing in $\mathbb{R}^+$ and injective; hence, $area(t)$ is an upper bound for the area of a connected configuration of S with $\phi(t)$ contacts. Observe that the *minimum value* for t is equal to the smallest semi-perimeter of a rectangle with area equal or greater than $A_{\min}$. The following lemma defines the formula to compute such a value:

Lemma 3. *Given $A \in N$, the smallest semi-perimeter of a rectangle with integer dimensions and area equal or greater than A is*

$$\left\lceil 2\sqrt{A} \right\rceil$$

Proof. Let $\sqrt{A} = n+\epsilon$, for $0 \le \epsilon < 1$. If $\epsilon = 0$, the smallest semi-perimeter of a rectangle with area equal or greater than n^2 is $2n$ and the claim follows. Otherwise, if $\epsilon > 0$, we have that the smallest semi-perimeter is equal or greater than $2n+1$, because $A > n^2 = area(2n)$. Moreover, observe that $area(2n+1) = n(n+1)$. Hence, we have that for $n^2 < A \le n(n+1)$ the smallest semi-perimeter is $2n+1$. With a similar reasoning it can be shown that, for $n(n+1) < A \le (n+1)^2$, the smallest semi-perimeter is $2n+2$. Now, observe that, for $0 < \epsilon \le 0.5$, we must have $n^2 < A \le n(n+1)$, since A is an integer and $(n+\epsilon)^2 < n(n+1)+1$. Hence, the function $\left\lceil 2\sqrt{A} \right\rceil$ maps correctly any $n^2 < A \le n(n+1)$ onto the value $2n+1$. Similarly, for $0.5 < \epsilon < 1$, we have $n(n+1) < A < (n+1)^2$ and the function $\left\lceil 2\sqrt{A} \right\rceil$ maps correctly any such area onto the value $2n+2$. □

From Lemma 3 it follows that the maximum number of contacts achievable in a connected configuration by a sequence with minimum area $A_{\min}$ is $\phi(\left\lceil 2\sqrt{A_{\min}} \right\rceil)$.

We now show that no non-connected configuration may exist with at least $\phi(\left\lceil 2\sqrt{A_{\min}} \right\rceil)$ contacts—this will conclude the proof on the new upper bound. Note that, by definition, a non-connected H-core must have at least one internal row or column, containing only non-singlet P beads or empty cells. Obviously, we dot not consider any border row or column because they would not belong to an H-core, by definition. Moreover, such a core can be always decomposed into two or more connected H-cores such that no HH contact spans two cores (as, otherwise, we could merge the two cores into a bigger connected core).

Lemma 4. *An HP sequence with minimum area $A_{\min}$ does not admit non-connected configurations with at least $\phi(\left\lceil 2\sqrt{A_{\min}} \right\rceil)$ contacts.*

Proof. We prove the result by contradiction. Assume that there exists one non-connected configuration with at least $\phi(\left\lceil 2\sqrt{A_{\min}} \right\rceil)$ contacts. Clearly, we must have $A_{\min} \ge 2$ for such a configuration to exist. We first prove it in the case when the configuration can be decomposed into two maximal connected cores. Let P_1 and P_2 be the semi-perimeters of such cores. By definition, the HH contacts must be fully contained in the two cores. Hence, it must hold that

$$\phi(P_1)+\phi(P_2) \ge \phi(\left\lceil 2\sqrt{A_{\min}} \right\rceil),$$

which can be easily proved to be equivalent to the condition

$$P_1+P_2 \le \left\lceil 2\sqrt{A_{\min}} \right\rceil\,.$$

Observe that the sum of the areas of the two connected cores must be at least $A_{\min}$. Hence, in the best-case the two areas are of the form B, $A_{\min}-B$ and their semi-perimeters are equal to $2\sqrt{B}$ and $2\sqrt{A_{\min}-B}$, respectively. The function $\sqrt{x}+\sqrt{A_{\min}-x}$, where $1 \le x < A_{\min}$, has two minima at $x=1$ and $x=A_{\min}-1$. Summing up, we obtain the following inequality

$$2\sqrt{1}+2\sqrt{A_{\min}-1} \le \left\lceil 2\sqrt{A_{\min}} \right\rceil < 2\sqrt{A_{\min}}+1,$$

which is true only for $A_{\min} < 2$, thus contradicting the hypothesis. Clearly, the proof also holds if the number of connected cores is greater than two. □

Let $C_{\text{area}} = \phi(\left\lceil 2\sqrt{A_{\min}} \right\rceil)$. By Lemmas 3 and 4 and the definition of ϕ it follows that C_{area} is *a new upper bound* for the number of contacts achievable by an HP sequence with minimum area $A_{\min}$. Indeed, no connected configuration with more than C_{area} contacts may exist, as the corresponding area would be smaller than $A_{\min}$, while, by Lemma 4, no non-connected configuration may exist with at least C_{area} contacts. Summing up, and remembering that $A_{\min} = m + |\mathcal{P}_S|$, we provide the following theorem stating the new bound:

Theorem 2. *The number of contacts achievable by an HP sequence with m H beads and corresponding sets $\mathcal{P}_S$ (P singlets) and $\mathcal{H}_I$ (internal H-H links) is at most equal to*

$$C_{area} = \phi(\left\lceil 2\sqrt{m+|\mathcal{P}_S|} \right\rceil) = 2m - |\mathcal{H}_I| - \left\lceil 2\sqrt{m+|\mathcal{P}_S|} \right\rceil$$

This represents *a new upper bound* on the number of contacts achievable by an HP sequence, in addition to the known upper bound presented in equation 3, and due to the parity issue.

Note that *either of the two upper bounds may be the most stringent*, depending on the sequence. Therefore, the new upper bound is the minimum of the two:

$$C_{\text{upper-bound}} = \min(C_{\text{area}}, C_{\text{parity}})$$

In section 5, Table 1 we show that the new bound outperforms the existing one in several cases.

4 A new algorithm for finding optimal structures in the 2-D lattice

We propose a new algorithm for the PSP problem based on the HP model on 2-D lattices. Our algorithm is based on a *pruned* exhaustive search. Given an HP sequence, the algorithm explores the corresponding tree representation of all its possible configurations, the size of the tree being exponential in the length of the sequence, in a top-down fashion. However, we add the ability to detect whether a partial configuration, corresponding to an internal node, may yield an optimal configuration. When this assertion is false, we say that the partial configuration is *prunable* and the algorithm can safely stop exploring the current branch of the tree, i.e., all the configurations that can be derived from (that start with) the current partial configuration are discarded.

Formally, the number of possible configurations on a 2-D lattice for an HP sequence of length n is at most 3^{n-2}, because, as explained in the previous section, we can assign a fixed position to the first two beads and place each of the remaining $n-2$ beads in at most three different ways. Hence, the set of all possible configurations of a sequence of length n can be represented using a complete 3-ary tree with height $n-2$. The number of iterations of the algorithm is therefore bounded by $(3^{n-1}-1)/2$ (the number of nodes of a 3-ary tree with height $n-2$). For each partial configuration of length k that is recognized as *prunable*, the number of iterations is reduced by a number bounded by $(3^{n-2-k}-1)/2$. The detection of *prunable* partial configurations is achieved using a number of pruning criteria, which exploit various properties of the state associated to a given partial configuration.

An example of search tree is depicted in Figure 2. Leaf nodes represent configurations; some of them are invalid as they are not self-avoiding walks, such as configuration URDL; others are valid but do not achieve the highest possible number of contacts, such as configuration UUUU; finally, some are valid and optimal, such as configurations URDD and URDR. Without actually reaching all leaves, the algorithm is able to compute if a partial configuration, such as for example UU, cannot possibly lead to an optimal configuration, and can prune computation early.

In Figure 3, we depict an example sequence of length 25, S_e= PHPH^2(PPH)^{2}P^5(HP)^{3}PHP, which will guide us, as a running example, through the algorithm. In our pictures H beads are depicted as squares, and they

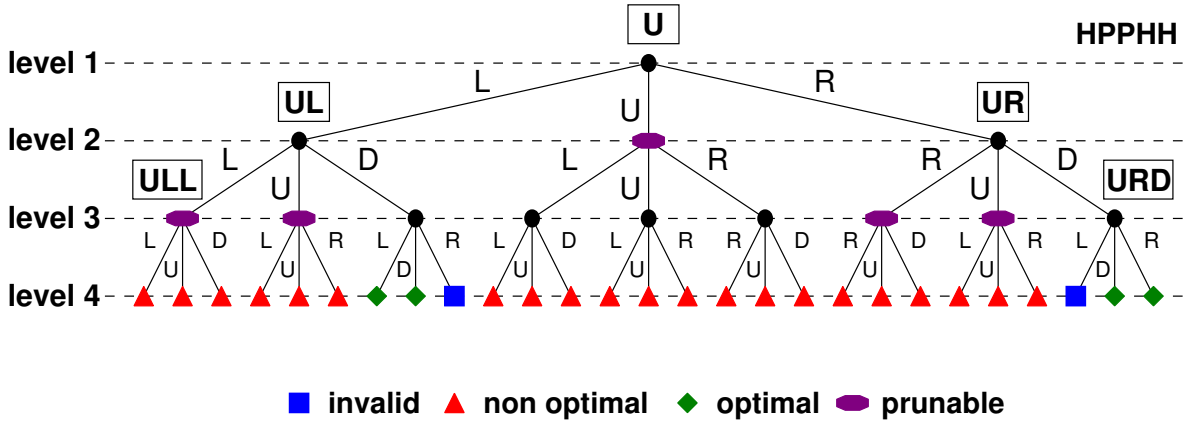

Figure 2: Search tree for sequence HPPHH. Level 0 is not represented, and the value of level 1, i.e., the direction of the second bead, is fixed at U, because the four branches of the root node are all equivalent up to rotations. Observe, furthermore, that all L-branches from the center, backbone of the tree are equivalent (specular) to their corresponding R-branches. Therefore, all corresponding calls are skipped. The left part of the tree is therefore not actually explored, and is shown here only for the sake of clarity.

are coloured in blue if they belong to $\mathcal{H}_E$, and in red if they belong to $\mathcal{H}_O$. P beads are not depicted, i.e., they appear as a straight line that connects the sequence. Above the sequence, the index of each bead is shown, from 1 to 25. There are 4 Hs of type blue, and 5 Hs of type red. Hence, we have $|\mathcal{H}_E| = 4$, $|\mathcal{H}_O| = 5$ and $C_{\text{parity}} = C_E = 8$. Moreover, we have that $|\mathcal{H}_I| = 1$ and $|\mathcal{P}_S| = 3$, so $C_{\text{area}} = 10$ and the stricter upper bound is given by C_{parity}. Consequently, in order to achieve 8 contacts, all 4 H beads belonging to $\mathcal{H}_E$ *must* achieve two contacts each. Observe that the energy of the optimal configuration(s) can be less than C_{parity}, i.e., there might be no configuration with such an energy. In the case of the example sequence mentioned above, an optimal configuration with C_{parity} contacts (8, in this case) does exist, and is pictured in Figure 4.

The main idea that we exploit in the proposed algorithm is based on the concept of **exposed site**. When a bead is placed on the lattice, it *exposes* sites where other beads *must* be located in order to create contacts. Hence, the placement of a bead imposes some constraints on a configuration, i.e., it reduces the degrees of freedom and, consequently, makes it possible to predict if a partial configuration may or may not lead to an optimal configuration. In particular, we define the notion of *weight* of an exposed site and show how to use the information about the number and weight of exposed sites to state whether a configuration cannot yield a given, sought-after number of contacts. Figure 5(a) shows a partial configuration of an HP sequence, where three sites are exposed by O-beads and two sites are exposed by E-beads.

Formally, we define *exposed site* an empty cell on the lattice that has at least one H bead as first-neighbour. We define the *weight* of an exposed site as the number of H beads that are first-neighbours of the site. The weight corresponds to the number of contacts that can be obtained by placing an H bead on the site. We say that an exposed site is of type O (E) if its first-neighbour(s) belong to the subset $\mathcal{H}_O$ ($\mathcal{H}_E$). Note that, analogously to the beads case, two exposed sites of the same type cannot be first neighbours.

We define the state of a partial configuration as the tuple

$$(C, (x_{\min}, y_{\min}), (x_{\max}, y_{\max}), W_O, W_E, S_O, S_E),$$

where

- C is the number of contacts achieved by the partial configuration.
- $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ are the coordinates of the lower left and upper right corner, respectively, of the smallest rectangle including all the H beads placed so far.
- W_O (W_E) is the number of *wasted* contacts, i.e., the sum of the weights of the exposed sites of type O (E) that have been occupied by a P bead.
- S_O (S_E) is the number of exposed sites of type O (E).

P	H	P	H	H	P	P	H	P	P	H	P	P	P	P	P	H	P	H	P	H	P	P	H	P
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Figure 3: (a) The sequence S_e= PHPH^2(PPH)^{2}P^5(HP)^{3}PHP, of length 25, and the corresponding indexes of each bead; (b) Graphical representation of S_e. The link between sequence-adjacent beads is shown as a black segment. H beads are depicted as squares, and they are coloured in blue if they belong to $\mathcal{H}_E$, and in red if they belong to $\mathcal{H}_O$. P beads are not depicted, i.e., they appear as a straight line that connects the sequence. There are 4 blue Hs in this sequence (at indexes 2, 4, 8, 24), and 5 red Hs (at indexes 5, 11, 17, 19, 21).

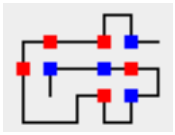

Figure 4: The single, optimal configuration for the example sequence S_e. It has C_{parity} contacts (8 in this case: 2 for each of the blue beads).

We denote as $h_E(k)$ the number of beads in $\mathcal{H}_E$ that occur in the suffix of length $n-k$ of the sequence, i.e.,

$$h_E(k) = |\{i \mid i > k \wedge i \in \mathcal{H}_E\}|. \tag{4}$$

$h_O(k)$ is defined analogously. For the example sequence, $h_E(15) = 1$, $h_O(15) = 3$(there is one remaining blue bead among the last 10 (25-15) beads, there are three remaining red beads among the last 10).

Our algorithm takes two input parameters in addition to the HP sequence. The first parameter is the number $C_I \leq \min(C_{\text{parity}}, C_{\text{area}})$ of expected contacts. Since the maximum number of contacts of the optimal configuration(s) is not known *a priori*, we parametrize it. The second parameter is the number $Area_I$, which is the maximum area of the H-core. The algorithm finds *all* the configurations of the sequence with a number of contacts equal at least to C_I and with H-core area equal at most to $Area_I$.

To find all the optimal configurations of a given sequence we proceed as follows. Given a sequence with contact bounds C_{parity} and C_{area}, we run the algorithm for each C_I value in the range $1, 2, \ldots, \min(C_{\text{parity}}, C_{\text{area}})$ in decreasing order, in such a way that we can stop as soon as at least one configuration with the expected energy is found. Concerning the area parameter, given the parameter C_I we have to find a bound on the maximum area of any configuration of the sequence with C_I contacts. Values of $Area_I$ below the maximum area are not safe as the algorithm could miss some or even all the configurations. Based on the results presented in Section 3, if $C_I = C_{\text{area}}$ we use equation 3 to compute the parameter $Area_I$ from the semi-perimeter $\left\lceil \sqrt{2A_{\min}} \right\rceil$, otherwise we set $Area_I$ to ∞, i.e., we disable the area pruning criteria.

While traversing the search tree, the algorithm recursively computes the state associated with the partial configuration corresponding to the current node in the tree. It then evaluates various *pruning* criteria that allow one to establish whether the current partial configuration *cannot* yield a configuration with C_I contacts.

The pruning criteria that we discuss first are based on the parameter C_I. We say that a waste of w contacts occurs if an exposed site, with weight w, is occupied by a P bead. The maximum number of contacts that are *exposed* by the H beads of type O is C_O. Since we want to find all the configurations with at least C_I contacts, we can then *waste* at most

$$W_O^{\max} = C_O - C_I$$

contacts exposed by H beads of type O. The same consideration holds for the H beads of type E. Note that, since $C_I \leq C_{\text{parity}} = \min(C_E, C_O)$, both $W_O^{\max}$ and $W_E^{\max}$ are non-negative and thus well defined. The pruning criterion is then:

Criterion 1. *If $W_E > W_E^{\max}$ or $W_O > W_O^{max}$, the current configuration is prunable.*

Figure 5(b) shows a partial configuration of the example sequence $S_e$in which one site of type E with unitary weight is wasted, i.e., $W_E = 1$. If $C_I = C_{\text{parity}}$, it follows that $W_E^{\max} = 0$ and thus the partial configuration is prunable, as it cannot possibly lead to an optimal configuration of C_{parity} contacts.

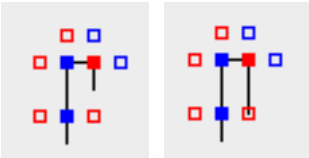

Figure 5: Partial configurations of the example sequence S_e: (a) Partial configuration UUURD. The H beads of types O and E are colored in red and blue, respectively. The link between sequence-adjacent beads is shown as a black segment. Empty squares are exposed sites; (b) Partial configuration UUURDD. The last placed P bead occupies a site exposed by the second bead and thus wastes a contact.

The next pruning criterion relates to exposed sites. Observe that the minimum number of exposed sites of type O that *must* be occupied, in order to *not* waste more than $W_O^{\max} - W_O$ contacts, is equal to

$$S_O - (W_O^{\max} - W_O)$$

because, in the most conservative case, each site has unitary weight and thus we can waste at most a number of sites equal to the number of wastable contacts; the same reasoning holds for the sites of type E. It follows that the number of H beads in the remaining suffix of the sequence (defined by the map $h_O(\cdot)$ in eq. 4) must be equal at least to the number of exposed sites that have to be occupied. Given a partial configuration of length k, the pruning criterion is then:

Criterion 2. *If $h_O(k) < S_E - (W_E^{\max} - W_E)$ or $h_E(k) < S_O - (W_O^{\max} - W_O)$, the current configuration is prunable.*

Figure 6(a) shows a partial configuration of the example sequence S_e in which there are not enough remaining beads of type O that may occupy the exposed sites of type E. Specifically, we have $h_O(8) = 4$, $S_E = 5$, and $W_E = 0$. Hence, if $C_I = C_{\text{parity}}$ the configuration is prunable.

We now focus on the number of contacts of a given configuration. In particular, we try to estimate the maximum number of remaining contacts achievable by a given partial configuration. If this number is smaller than $C_I - C$, (C is the current number of contacts of the partial configuration) we may conclude that the current configuration cannot yield C_I contacts. A pruning criterion follows:

Criterion 3. *Let C_{left} be the maximum number of remaining contacts that can be achieved by the current configuration. If $C_{left} < (C_I - C)$, the current configuration is prunable.*

We show next how to estimate C_{left}. The first observation concerns the number of contacts that an H bead generates when placed on the lattice. By Lemma 1, any internal bead can make at most 2 contacts while the last bead can make at most 3 contacts. Moreover, note that the last bead can make 3 contacts only if it is an H bead. Obviously, the placement of the first bead does not give rise to any contact. This fact leads to a simple loose estimate of the maximum number of contacts achievable by a partial configuration of length k:

$$2(h_E(k) + h_O(k)) + e_n$$

where e_n is 1 if $\hat{a}_n = H$ and 0 otherwise. However, it is possible to obtain a closer (i.e., correct, but less conservative) estimate by exploiting the information on the number and weight of the existing exposed sites. As explained above, an H bead, when placed on the lattice, exposes either two sites, if it is internal, or three if it is the first or the last bead; in order to generate contacts the exposed sites must be occupied by a bead of opposite type. There are three main conditions under which an exposed site cannot be occupied by such a bead with a resulting waste equal to the corresponding weight:

1. the site is occupied by a P bead;
2. there are not as many beads of opposite type as the available sites;
3. the site is unreachable.

Figures 6(a) and 6(b) show examples of conditions 2 and 3. Let $1 \leq k < n$ be the length of a given partial configuration. We focus on the beads of type O, but the reasoning is analogous for the other case. A crucial

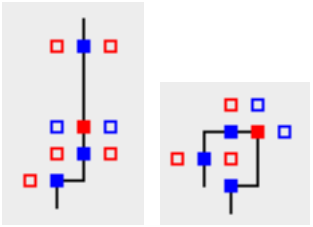

Figure 6: Partial configurations of the example sequence S_e: (a) Partial configuration URUUUUUU. There are four beads of type O left; hence, it is not possible to occupy all the sites exposed by the beads of type E (there are 5 of them). This situation will result necessarily in wasting at least a contact, and therefore C_{parity} cannot be achieved. Hence this configuration is prunable; (b) Partial configuration UURRDDLD showing an unreachable exposed site with weight 3.

observation is that the beads cannot make more than $2|\mathcal{H}_O|+e_1+e_O$ contacts (see eq. 1), which implies that we can estimate the number of potential residual contacts by considering the H beads and the exposed sites of type O only. We can write the maximum number of remaining contacts as C_1+C_2, where C_1 and C_2 are the contacts achievable by the remaining and placed beads, respectively. As for the beads that have not been placed yet we assume that the most favourable case applies, i.e., $C_1=2h_O(k)+e_O$. Let $s_O(i)$ be the number of exposed sites of type O with weight i, for $i=1,\dots,4$. s_E is defined analogously. Concerning the placed beads, the respective number of potential contacts is $2(|\mathcal{H}_O|-h_O(k))+e_1$. However, the following inequality holds:

$$C_2 \leq \sum_{i=1}^{4} i\cdot s_O(i) \leq 2(|\mathcal{H}_O|-h_O(k))+e_1-C\,,$$

i.e., the weighted sum of the exposed sites is the maximum number of contacts still achievable by the placed beads and it cannot exceed the number of potential contacts minus the existing contacts. Hence, we can use $C_2=\sum_{i=1}^{4} i\cdot s_O(i)$. Observe that the number of contacts wasted because of condition (1), i.e., W_O, is equal to the third term minus the second term. When such two quantities are equal, no site is wasted; instead, if the difference is positive, one or more sites are wasted, because, when a site is occupied by a P bead, the number of exposed sites decreases while C does not increase. To take into account conditions (2) and (3) we need a more tight bound for the term $\sum_{i=1}^{4} i\cdot s_O(i)$. Observe that the sites with weight 4 cannot be reached and only one site with weight 3 can be occupied (by the last bead, provided that it is an H bead of type E). Moreover, note that we cannot occupy more than $h_E(k)$ sites, as we would not have enough beads available. In order to account for the most conservative scenario, we assume that the sites with highest weights are occupied. The number of sites that can be effectively occupied is then $M_O=M_O^3+M_O^2+M_O^1\leq S_O$, where

$$\begin{aligned} M_O^3 &= \min(s_O(3), e_E)\\ M_O^2 &= \min(s_O(2), h_E(k)-M_O^3)\\ M_O^1 &= \min(s_O(1), h_E(k)-M_O^3-M_O^2) \end{aligned}$$

Hence, if $M_O<S_O$, we waste $\sum_{i=0}^{4} i(s_O(i)-M_O^i)$ contacts because of conditions (2) and/or (3). Summing up, the resulting formula is:

$$2h_O(k)+e_O+3M_O^3+2M_O^2+M_O^1$$

The next pruning criterion concerns the H-core, i.e., the smallest rectangle containing all the H beads placed in the current configuration:

Criterion 4. *If the area of the H-core encoded by the coordinates* $(x_{\min},y_{\min}),(x_{\max},y_{\max})$ *is bigger than* $Area_I$ *the current configuration is prunable.*

where we recall that $Area_I$ is the parameter of the algorithm that specifies the maximum area of the H-core. Observe that the parameter $Area_I$ is constrained by the inequality $Area_I\geq A_{\min}$. Moreover, note that the maximum number of non-singlets P beads in the H-core is bounded by $Area_I-A_{\min}$, as, otherwise, the H-core could not contain all the H beads. This observation leads to another pruning criterion:

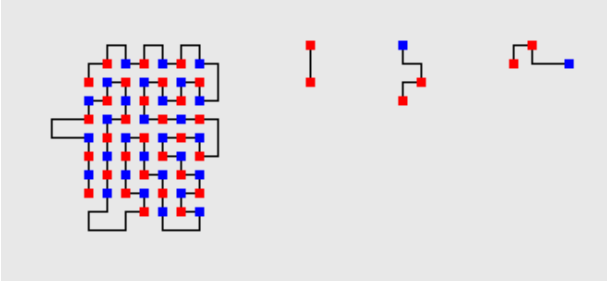

Figure 7: Example of how early in the search Criterium 6 is able to prune. The picture shows an optimal configuration for sequence $S1-9$ (reversed), and three partial configurations that are pruned because of Criterium 6. The first partial configuration is pruned as early as the third bead, and the second and third configuration as early as the fifth bead.

Criterion 5. *If the number of non-singlets P beads in the current H-core is greater than* $Area_I - A_{\min}$*, the current configuration is prunable.*

When $C_I = C_{\text{area}}$ we can also devise another pruning criterion, based on Theorem 1. By Theorem 1 and the definition of ϕ it follows that in any connected H-core with dimensions (L_1, L_2), m H beads, and $\phi(L_1+L_2)$ contacts the waste must be *exactly* $2(L_1L_2 - m)$, i.e.,

$$\sum_{i\in R_P}(n_H(i,*)-1)+\sum_{j\in C_P}(n_H(*,j)-1)=0\,,$$

where we recall that n_H is the number of H beads in a row (column) after collapsing all the sequences of adjacent H beads into a single H bead. Moreover, it is a simple observation that in a configuration with semiperimeter $\left\lceil 2\sqrt{A_{\min}}\right\rceil$, which is the smallest semiperimeter possible to contain the whole sequence, we have $n_H \geq 1$ for any row and column, i.e., each row and column contains at least one H bead. This observation leads to our final pruning criterion:

Criterion 6. *If* $C_I = C_{area}$ *and there is one row or column with* $n_H > 1$ *the current configuration is prunable.*

The criterium in practice says that, in a configuration with C_{area} contacts, there cannot be in any line or in any column a segment of Ps delimited by Hs at each end. Figure 7 exemplifies this concept: a benchmark sequence is pictured, $S1-9$, with one of its optimal configurations. The picture then shows three partial configurations that are pruned because of Criterium 6. In fact, in all three cases we have a segment of Ps delimited by Hs: in the first and second case this happens along a column, while in the third case it happens along a row.

One important thing to notice is how early in the search this criterium is able to prune: already in the third or fifth bead, in a sequence of length 64. Notice that, instead, the criteria based on exposed sites would not be effective, in this case, this early in the search. Criterium 6, in combination with Criterium 5, is indeed the main reason for MINWALK to perform so effectively, in all those benchmark sequences where the optimal number of contacts is equal to the upper bound C_{area}.

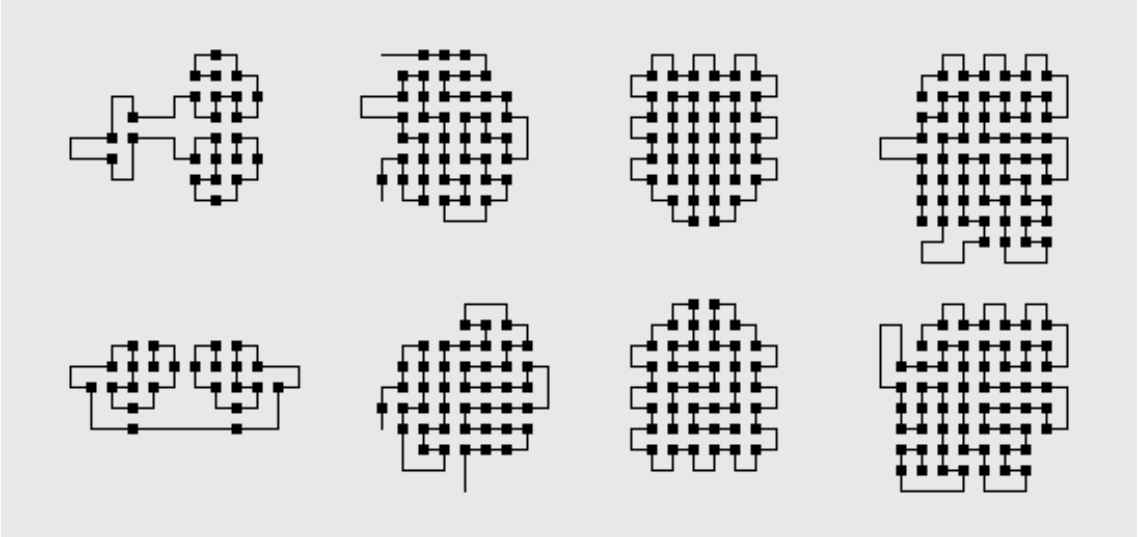

Figure 8: Two optimal configurations for each of the sequences S1-6, S1-7, S1-8, and S1-9.

5 Experimental results

In this section we present some experimental results on the new devised bound, and on our algorithm, named MINWALK.

The algorithm was implemented in the C programming language and running times have been measured using the **clock** function. The tests were performed on a 2.53 GHz Intel Xeon E5630 machine with Ubuntu 10.04 and `GNU C Compiler 4.4`.

We tested MINWALK on various benchmarks. The first experiment consisted in running MINWALK on the first nine sequences of the standard HP benchmarks [1]. The sequences are shown in Table 1, together with the associated contact bounds C_{parity}, C_{area} and C_{LP} which corresponds to the value presented in [1]. As can be seen from the reported values, our new bound C_{area} is better than the best known one in six cases, and in three cases it is also equal to the real best energy value. Moreover, observe that the contact bounds reported by [1] are always equal to C_{parity}.

Table 2 shows the experimental results for this benchmark. In particular, we reported for each sequence the running time, the best energy value found and the corresponding number of optimal configurations. This table also includes the running times, taken from the benchmark in [15], of the best non exact algorithms, namely PERM [5], F&F [12], HPPFP-3 [14] and PFold-P [15]. Note that these algorithms find a single configuration, as opposed to our algorithm which finds all the optimal configurations of a given sequence. To provide a fair comparison we also show in this table the running time of MINWALK to compute the first optimal configuration found. The experimental results show that our algorithm is fast and competitive if compared with the best non exact methods. The only exception is sequence S1-6, which requires a huge amount of time to be solved with our method. One possible explanation for this fact is the large difference (5) between the optimal energy for this sequence, and its bound C_{area}. Therefore, in this case we perform 5 runs of MINWALK (with C_I from 25 to 21) on an input of length 50 with the H-core pruning criteria disabled. Concerning the other sequences, our algorithm finds the first optimal configuration in the order of seconds for all of them, and it is always as fast or faster than all other algorithms in the state of the art. In normal mode, except for sequences S1-7 and S1-9 which require more time, our algorithm finds all the configurations of the sequences also in the order of seconds.

For example, for sequence S1-5, all 492 optimal configurations (which we know to be optimal, by the results in Table 2) are found in a fraction of a second.

The results obtained for the long sequences S1-7 and S1-9 are also good. Our running times are in the order of minutes and are better than the ones obtained by the ACO algorithms HPPFP-3 and PFold-P, while the PERM and F&F algorithms are faster. However, we recall that MINWALK in this mode finds *all* the optimal configurations of the sequence, which in the case of S1-9 are a huge number (7710). Sequence S1-8 represents

[1] http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html

ID	Length	C_{NEW}	C_{LP}	C_{parity}	C_{area}	E*	Sequence
S1-1	20	**10**	11	11	10	9	$(HP)^2PH^2PHP^2HPH^2P^2HPH$
S1-2	24	**11**	**11**	11	11	9	$H^2(P^2H)^7H$
S1-3	25	**8**	**8**	8	8	8	$P^2HP^2(H^2P^4)^3H^2$
S1-4	36	**14**	16	16	14	14	$P^3H^2P^2H^2P^5H^7P^2H^2P^4H^2P^2HP^2$
S1-5	48	**23**	25	25	23	23	$P^2H(P^2H^2)^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$
S1-6	50	**25**	**25**	25	28	21	$H^2(PH)^3PH^4PH(P^3H)^2P^4H(P^3H)^2PH^4P(HP)^3H^2$
S1-7	60	**37**	40	40	37	36	$P^2H^3PH^8P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$
S1-8	64	**42**	43	43	42	39	$H^{12}(PH)^2(P^2H^2)^2P^2HP^2H^2PPH^2P^2HP^2(H^2P^2)^2(HP)^2H^{12}$
S1-9	85	**53**	-	58	53	53	$H^4P^4H^{12}P^6(H^{12}P^3)^3HP^2(H^2P^2)^2HPH$

Table 1: Various sequences from the standard HP benchmarks and corresponding length, contact bounds and best energy values. The new bound C_{NEW}, equal to the minimum between C_{parity} and C_{area}, improves in several cases on the state of the art C_{LP} (which is, at least for these sequences, effectively equal to C_{parity}).

Sequence	MINWALK	MINWALK (single)	PERM	F&F	HPPFP-3	PFold-P	Energy	Degeneracy
S1-1	0.01s	0.01s	<1s	0s	<1s	0.06s	**9**	2
S1-2	0.03s	0.01s	<1s	2s	<1s	0.4s	**9**	19
S1-3	0.01s	0.01s	2s	0.5s	<1s	0.2s	**8**	16
S1-4	0.01s	0.01s	<1s	4s	4s	1.1s	**14**	192
S1-5	0.5s	0.01s	2s	10s	1m	13.3s	**23**	492
S1-6	85h	38h	3s	22s	15s	15.4s	**21**	897
S1-7	3.5m	0.4s	4s	56s	20m	4m	**36**	242
S1-8	0.3s	0.01s	78h	24s	1.5h	35m	**42**	39
S1-9	14m	30s	60s	1.3m	24h	4.5h	**53**	7710

Table 2: Experimental results of the algorithm on various biological sequences from the standard HP benchmark, of lengths up to 85.

an interesting case. The optimal energy value of -42 is found, with its 39 degenerate versions, in less than a second. Only the F&F algorithm is able to find a solution in the order seconds; the running time of the other non exact algorithms on this sequence is in the order of hours and even days in the worst-cases.

Figure 8 depicts sample optimal configurations for S1-6, S1-7, S1-8, and S1-9. Note that MINWALK is able to guarantee the optimality and the degeneracy of the solutions found.

Table 3 shows the the experimental results for various biological and random sequences of length $\sim$30 taken from the supplementary material of Ref. [14]. Our algorithm performed well also in this experiment; the running time is in the order of seconds on average and it is less than two minutes in the worst-case.

We also tested MINWALK on random sequences of length 45 (reported in Table 5); Table 4 again shows the running time, the best energy value found and the corresponding number of optimal configurations. There is no algorithm in the state of the art that can return optimal configurations in the 2-D lattice for input sequences of this size. MINWALK does so in a matter of hundreds of seconds.

Last, we noted that the optimal configurations of a consistent number of standard benchmarks sequences have an H-core area equal, or very close to, A_{min} (recall that A_{min} is the number of Hs+P-singlets in the sequence, and achieving A_{min} basically corresponds to packing all Hs and P-singlets in a compact H-core).

6 Conclusions

We have presented new results on HP sequences and configurations in 2-D lattice and a novel algorithm to solve the Protein Structure Prediction problem in the HP model in 2-D lattice. In particular, we discovered a new relationship between the energy and the area of a specific type of native configurations and, on top of this result, a *new upper bound* on the energy achievable by an HP sequence. The presented algorithm is fast and competitive even if compared with the speed of non exact methods, and *guarantees the optimality* of the solutions found. It is also the first exact algorithm able to solve random instances of length up to 45, and specific benchmarks instances of length up to 85. In the future we aim to extend our algorithm to the 3-D

Sequence	Time (sec)	Energy	Degeneracy
B30-1	105.08	12	104
B30-2	1.22	10	860083
B30-3	0.01	14	6
B30-4	4.88	10	265
B30-5	3.55	13	243
B30-6	3.59	13	7
B30-7	61.23	16	84
B30-8	2.53	8	494
B30-9	0.39	18	401
B30-10	31.48	16	307
R30-1	0.31	9	1595
R30-2	0.12	13	137
R30-3	0.13	10	1964
R30-4	5.72	9	2606
R30-5	1.04	13	10
R30-6	0.11	11	6
R30-7	1.69	8	15914
R30-8	0.13	12	50
R30-9	10.85	9	7167
R30-10	55.41	12	74

Table 3: Experimental results of the algorithm on various biological and random sequences from [14]. All these sequences have a length of around 30.

Sequence	Time (sec)	Energy	Degeneracy
R45-1	95.1	24	8
R45-2	68.68	17	76
R45-3	113.02	16	161
R45-4	28.52	18	4
R45-5	2.45	22	2
R45-6	563.78	17	259675
R45-7	767.35	25	190
R45-8	54.84	22	120
R45-9	158.94	21	77
R45-10	248.39	22	4

Table 4: Experimental results of the algorithm on random sequences of length 45.

ID	Sequence
R45-1	HPHPHHHHHHHPHPPHHPPHHPHHHHPPHHHHPPHPHHPPPPPHHH
R45-2	PHHPPPPHPPHHHPHPHHPPHHHHHPPHPPPHPPHPHPPPPHPPH
R45-3	HHPHHPHPHPPPHHPPPHHPHHPHPHPHPPPPPPPPPHPPPPHHH
R45-4	PPHHHPPHPPPPPPPHPHPHHHHHHHHHPHPHPHPPPPPPPHPPPHH
R45-5	HPHHHHHHHHHHHPHPHPPPPHHHPPHPHPPHPHPPPPHPPHPHHP
R45-6	PPPHHHPPHPHHPPHHHHPHHHPPHPHHPHHPHHHHHPPPPPPPPP
R45-7	HHPPPHHHPHPPHHHHHHHHHPHHHHPHPHPHPPHPPHHPHPHHHH
R45-8	HHPPHPPHHHPHHHHPPHHHHHHHHPPPPPPPPPHPHPPHHPHPPH
R45-9	HHPPHHPHHHPHHHPHHHHPPHPHPHPPHHPPPPPHPHHHPHPPPH
R45-10	HHPPPPHHPPHHHHHPHPHHPHPHHPPHPPPPHHHHPHHHHPPHP

Table 5: Random HP sequences of length 45.

lattice case, and compare it in performance with [2].

References

[1] N. Ahn and S. Park. Finding an upper bound for the number of contacts in hydrophobic-hydrophilic protein structure prediction model. *Journal of Computational Biology*, 17(4):647–656, 2010.

[2] R. Backofen and S. Will. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. *Journal of Constraints*, 11(1):5–30, Jan. 2006.

[3] B. Berger and F. T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[4] P. Crescenzi, D. Goldman, C. H. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3):423–466, 1998.

[5] H.-P. Hsu, V. Mehra, W. Nadler, and P. Grassberger. Growth algorithms for lattice heteropolymers at low temperatures. *Journal of Chemical Physics*, 118(1):444–451, 2003.

[6] A. Irback and C. Troein. Enumerating designing sequences in the HP model. *Journal of Biological Physics*, 28(1):1–15, Jan. 2002.

[7] S. Istrail and F. Lam. Combinatorial algorithms for protein folding in lattice models: A survey of mathematical results. 2009.

[8] K. F. Lau and K. A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989.

[9] N. Lesh, M. Mitzenmacher, and S. Whitesides. A complete and effective move set for simplified protein folding. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, RECOMB '03, pages 188–195, New York, NY, USA, 2003. ACM.

[10] F. Liang and W. H. Wong. Evolutionary monte carlo for protein folding simulations. *Journal of Chemical Physics*, 115(7):3374–3380, 2001.

[11] M. Mann, D. Maticzka, R. Saunders, and R. Backofen. Classifying protein-like sequences in arbitrary lattice protein models using LatPack. *HFSP Journal*, 2(6):396–404, 2008.

[12] C. Rego, H. Li, and F. Glover. A filter-and-fan approach to the 2d hp model of the protein folding problem. *Annals OR*, 188(1):389–414, 2011.

[13] V. Shahrezaei and M. R. Ejtehadi. Geometry selects highly designable structures. *Journal of Chemical Physics*, 113(15):6437–6442, 2000.

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

[15] T. Thalheim, D. Merkle, and M. Middendorf. Protein folding in the hp-model solved with a hybrid population based aco algorithm. *International Journal of Computer Science*, 35(3):291–300, 2008.

[16] R. Unger and J. Moult. A genetic algorithm for three dimensional protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588. Morgan Kaufmann, 1993.