A robust class of stable proteins in the 2D HPC model

Alireza Hadj Khodabakhshi, Ján Maňuch, Arash Rafiey, and Arvind Gupta

School of Computing Science
8888 University Drive, Simon Fraser University
Burnaby, BC, V5A 1S6, Canada
alireza@cs.sfu.ca, jmanuch@sfu.ca, arashr@cs.sfu.ca, arvind@mitacs.ca

Abstract. The inverse protein folding problem is that of designing an amino acid sequence which has a prescribed native protein fold. This problem arises in drug design where a particular structure is necessary to ensure proper protein-protein interactions. The input to the inverse protein folding problem is a shape and the goal is to design a protein sequence with a unique native fold that closely approximates the input shape. Gupta *et al.* [1] introduced a design in the 2D HP model of Dill that can be used to approximate any given (2D) shape. They conjectured that the protein sequences of their design are stable but only proved the stability for an infinite class of very basic structures. In [2], we have introduced a refinement of the HP model, in which the cysteine and non-cysteine hydrophobic monomers are distinguished and SS-bridges which two cysteines can form are taken into account in the energy function. This model was called the 2D HPC model. In [2], the snake structures in the HPC model were introduced and it was conjectured that they are stable. In this paper, we show that this conjecture is true for a subclass of snake structures. This subclass is robust enough to approximate any given 2D shape, although more coarsely than the general constructible structures proposed in [1]. In the proof we use a semi-automated tool 2DHPSolver developed in [2].

1 Introduction

It has long been known that protein interactions depend on their native three-dimensional fold and understanding the processes and determining these folds is a long standing problem in molecular biology. Naturally occurring proteins fold so as to minimize total free energy. However, it is not known how a protein can choose the minimum energy fold amongst all possible folds [3].

Many forces act on the protein which contribute to changes in free energy including hydrogen bonding, van der Waals interactions, intrinsic propensities, ion pairing, disulfide bridges and hydrophobic interactions. Of these, the most significant is hydrophobic interaction [4]. This led Dill to introduce the *Hydrophobic-Polar model* [5]. Here the 20 amino acids from which proteins are formed are replaced by two types of monomers: hydrophobic (H or '1') or polar (P or '0')

depending on their affinity to water. To simplify the problem, the protein is laid out on vertices of a lattice with each monomer occupying exactly one vertex and neighboring monomers occupy neighboring vertices. The free energy is minimized when the maximum number of hydrophobic monomers are adjacent in the lattice. Therefore, the "native" folds are those with the maximum number of such HH contacts. Even though the HP model is the simplest model of the protein folding process, computationally it is an NP-hard problem for both the two-dimensional [6] and the three-dimensional [7] square lattices.

In many applications such as drug design, we are interested in the complement problem to protein folding: *inverse protein folding* or *protein design*. The *inverse protein folding problem* involves starting with a prescribed target fold or structure and designing an amino acid sequence whose native fold is the target (positive design). A major challenge in designing proteins that attain a specific native fold is to avoid proteins that have multiple native folds (negative design). We say that a protein is *stable* if its native fold is unique. In Gupta *et al.* [1], a design in the 2D HP model that can be used to approximate any given (2D) shape was introduced and it was shown that approximated structures are native for designed proteins (positive design). It was conjectured that the protein sequences of their designed structures are also stable but only proved for an infinite class of very basic structures (arbitrary long "I" and "L" shapes), as well as computationally tested for over 48,000 structures (including all with up to 9 tiles). Design of stable proteins of arbitrary lengths in the HP model was also studied by Aichholzer *et al.* [8] (for 2D square lattice) and by Li *et al.* [9] (for 2D triangular lattice), motivated by a popular paper of Brian Hayes [10].

In natural proteins, sulfide bridges between two cysteine monomers play an important role in improving stability of the protein structure [11]. In our previous work [2] we extended the HP model by adding the third type of monomers, cysteines, and incorporating sulfide bridges between two cysteines into energy model. This model is called the HPC (hydrophobic-polar-cysteine) model. The cysteine monomers in the HPC model act as hydrophobic, but in addition two neighboring cysteines can form a sulfide-sulfide bridge to further reduce the energy of the fold. Therefore, between many folds of the same protein with the same number of hydrophobic bonds the one with the maximum number of sulfide bridges is the most stable fold. This added level of stability can help in proving formally that the designed proteins are indeed stable.

In [2] we introduced a class of structures called the *snake structures*. The class of snake structures is a subset of the class *linear structures* introduced by Gupta *et al.* [1]. The linear structures are formed by a sequence of "plus" shape tiles, cf. Figure 1(a), connected by overlapping two pairs of polar monomers (each coming from a different tile). The structures are linear which means that every tile except the first and the last is attached to exactly two other tiles. In the snake structures every second tile is a bending tile. The first, last and the bending tiles in a snake structure contain cysteine monomers while the rest of the tiles contain hydrophobic monomers. In [2] we conjectured that the protein of snake structures are stable and we proved it under an additional assumption

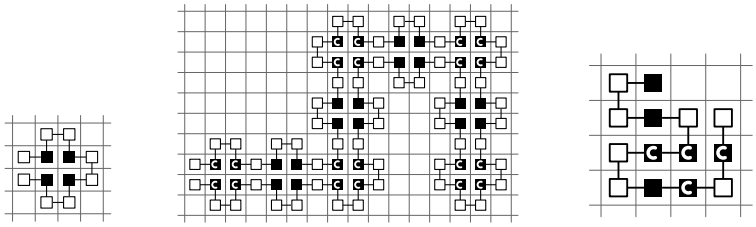

Fig. 1. (a) The basic building tile for constructible structures: black squares represent hydrophobic and white polar monomers. The lines between boxes represent the peptide bonds between consecutive monomers in the protein string. (b) An example of snake structure. The bending tiles use cysteines (black squares marked with C). (c) Example of energy calculation of a fold in HPC model. There are 5 contacts between hydrophobic monomers, thus the contact energy is -5. There are three potential sulfide bridges sharing a common vertex, hence only one can be used in the maximum matching. Thus the sulfide bridge energy is -2 and the total energy is -7.

that non-cysteine hydrophobic monomers act as cysteine ones, i.e., they tend to form their own bridges to reduce the energy. This model was called the strong HPC mode. Even though this model is artificial, we used it to demonstrate that our techniques can be used to prove stability of snake structures in the "proper" HPC model.

In this paper we consider a subclass of snake structures which is robust enough to that are restricted enough to approximate any given shape and the same time restricted enough to be proved stable using our techniques. We call this subclass *wave structures*. The wave structure are instances of the snake structures that do not contain any occurrence of the four forbidden motifs in Figure 2. We believe this a first robust design formally provable that it is stable.

This paper is organized as follows. We start by the definition of the HPC model and introducing the wave structures in Section 2. In Section 3 we explain our proof techniques and used them to prove the protein of any wave structure is stable.

2 Definitions

In this section we define the HPC model introduced in [2] as extension of the HP model of Dill [5] and introduce wave structures.

2.1 Hydrophobic-polar-cysteine (HPC) model

Proteins are chains of 20 types of amino acids. In the HPC model, we consider only 3 types of amino acids: polar, cysteine and non-cysteine hydrophobic. We can represent a protein chain as a string $p = p_1 p_2 \dots p_{|p|}$ in $\{0, 1, 2\}^*$, where "0"

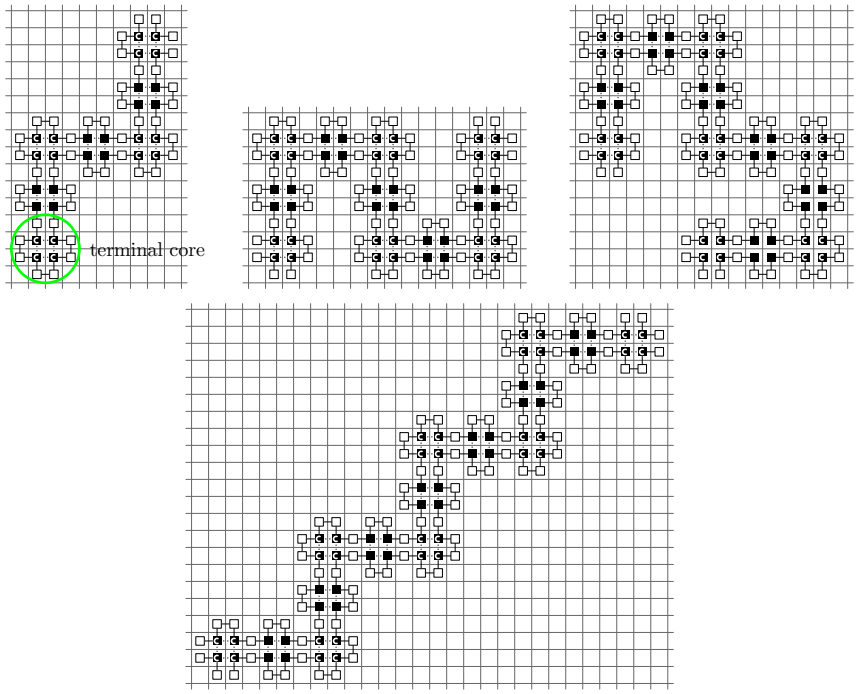

Fig. 2. Forbidden motifs in wave structures.

represents a polar monomer, "1" a hydrophobic non-cysteine monomer and "2" a cysteine monomer.

The proteins are folded onto the regular lattice. A *fold* of a protein p is embedding of a path of length n into lattice, i.e., vertices of the path are mapped into distinct lattice vertices and two consecutive vertices of the path are mapped to lattice vertices connected by an edge (a peptide bond). In this paper we use the 2D square lattice.

A protein will fold into a fold with the minimum free energy, also called a *native fold*. In the HP model only hydrophobic interactions between two adjacent hydrophobic monomers which are not consecutive in the protein sequence (*contacts*) are considered in the energy model, with each contact contributing with -1 to the total energy. In addition, in the HPC model, two adjacent non-consecutive cysteines can form a sulfide bridge contributing with -2 to the total energy. However, each cysteine can be involved in at most one sulfide bridge. More formally, any two adjacent non-consecutive hydrophobic monomers (cysteine or non-cysteine) form a contact and the contact energy is equal to -1 times the number of contacts; and any two adjacent non-consecutive cysteines form a *potential* sulfide bridge and the sulfide-bridge energy is equal to -2 times the number of matches in the maximum matching in the graph of potential sulfide bridges. The total energy is equal to the sum of the contact and sulfide bridge energies. For example, the energy of the fold in Figure 1(c) is $(-5)+(-2)=-7$. (Note that the results in the paper are independent on the exact value of the energy of sulfide bridge, as long as it is negative, and therefore we did not research on determination of the correct value for this energy.)

There might be several native folds for a given protein. A protein with a unique native fold is called *stable* protein.

2.2 Wave structures

In Gupta *et al.* [1], a wide class of 2D structures, called *constructible structures*, was introduced. They are formed by a sequence of "plus" shape tiles, cf. Figure 1(a), connected by overlapping two pairs of polar monomers (each coming from different tile). It was conjectured that these structures are stable and proved for two very simple subclasses of the linear structures, namely for L_0 and L_1 structures. The L_0 and L_1 structures consist of an arbitrary large sequence of tiles in the shape of a straight line and the letter L, respectively. Note that although L_1 structures are still quite simple, the proof of their stability involves analysis of a large number of cases. In our previous work [2], we introduced a subclass of constructible structures, snake structures, and refine it for the HPC model with nice combinatorial properties, e.g., in the proteins of such structures any two consecutive hydrophobic monomers are of the same type if there two polar monomers between them and are of different type if there is one polar monomer between them. This significantly reduces the case analysis and we conjectured that the snake structures are stable.

The snake structures are *linear* structures which means that every tile t_i except the first t_1 and the last t_n is attached to exactly two other tiles t_{i-1}

and t_{i+1} (and the first and the last ones are attached to only one tile, t_2 and t_{n-1}, respectively). In addition, in a snake structure the sequence of tiles has to change direction ("bend") in every odd tile. The hydrophobic monomers of these "bending" tiles are set to be cysteines, and all other hydrophobic monomers are non-cysteines, cf. Figure 1(b). Although, the snake structures are more restricted, the proof of their stability under the HPC model required the analysis of huge number of cases. However, in [2] we were able to prove that they are stable under the artificial strong HPC model. This model assumes that the non-cysteine hydrophobic monomers form SS-bridges of their own to reduce the energy of the conformation. Notice that cysteine and none-cysteine monomers cannot form SS-bridges. Although, the strong HPC model is not a proper biological model, the proof of the stability of the snake structures under the strong HPC model raised the hope for finding the structures that can be proved to be stable under the proper HPC model.

In this paper, we introduce a subclass of the snake structures called the *wave structures* and formally prove that they are stable under the proper HPC model. Although, the wave structures is only a subclass of the snake structures they can still approximate any given shape in 2D square lattice. The wave structure are instances of the snake structures that do not contain occurrence of the four forbidden motifs in Figure 2. The wave structures can be constructed using a set of four super-tiles and their flipped versions (cf. Figure 3).

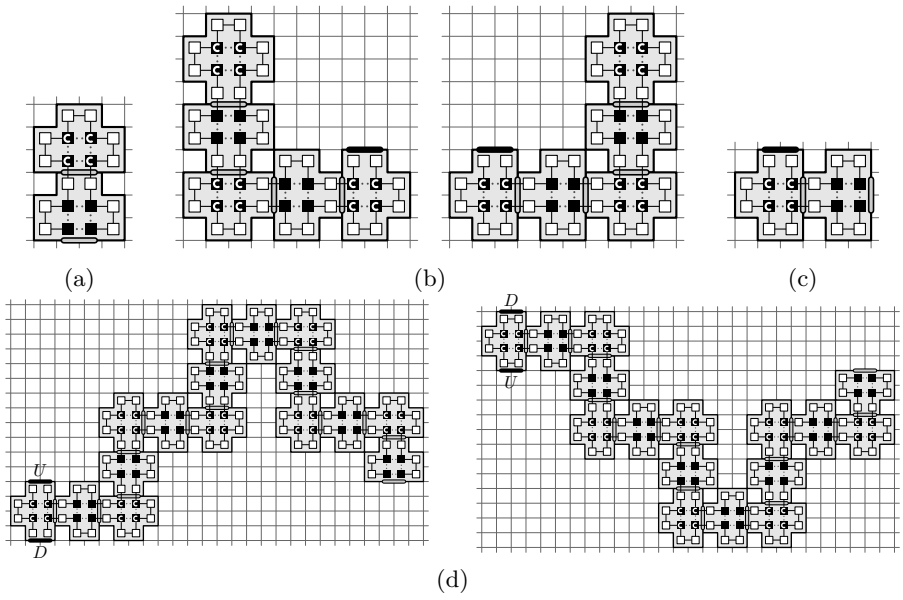

Fig. 3. Super-tiles used to construct wave structures: (a) starting super-tile; (b) unflipped and flipped versions of terminating super-tile; (c) bending super-tile; and (d) flipped and non-flipped versions of regular tile.

The super-tiles are simple instances of the constructible structures. The *starting* super-tile has one receptor and consists of two basic tiles (Figure 3(a)), the *terminating* super-tile has one ligand and consists of 5 basic tiles (Figure 3(b)), the *bending* super-tile has one ligand and one receptor and consists of two tiles (Figure 3(c)), and the *regular* super-tile has two ligands and one receptor and consists of 16 basic tiles (Figure 3(d)). The receptor of one super-tile can connect to the ligand of another one however, the regular super-tile must only connect through one of its ligands. A wave structure is a partial tiling of the two-dimensional grid obtained by the following procedure.

1. Place the starting super-tile into the grid and place a regular super-tile into the grid so that its U ligand is attached to the receptor of the staring gadget.
2. Let the last placed super-tile be a (flipped) regular super-tile R; either place a (flipped) regular super-tile so that its U ligand is attached to the receptor of R and continue with step 4 or place a bending super-tile such that its ligand is attached to receptor of R and continue with step 3.
3. Let the last placed super-tile be a bending super-tile B and let R be a regular super-tile attached to B. If R is a flipped super-tile then attach a new non-flipped regular super-tile to B otherwise, attach a new flipped super-tile to B. The new super-tile can be attached either with U or D ligand depending on intended direction of the bend.
4. Continue with step 2 or end the structure by attaching a (flipped) terminating super-tile to the last placed (flipped) regular super-tile.

In the above procedure the super-tiles are placed into the grid such that they do not overlap. An example of a wave structure is depicted in Figure 4.

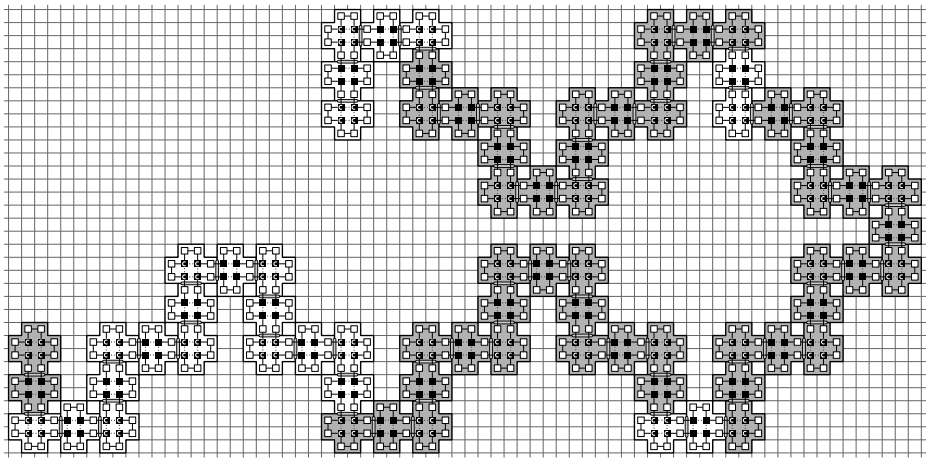

Fig. 4. An example of a wave structure. It consists of 8 super-tiles. The borders between super-tiles are marked by the change of underlying color of the core tiles.

As observed in [2] for snake structures, approximately 40% of all monomers in wave structures are hydrophobic and half of those are cysteines. Thus approximately 20% of all monomers are cysteines. Although, the most of naturally occurring proteins have much smaller frequency of cysteines, there are some with the same or even higher ratios: 1EZG (antifreeze protein from the beetle [12]) with 19.5% ratio of cysteines and the protein isolated from the chorion of the domesticated silkmoth [13] with 30% ratio.

Note that the wave structures can still approximate any given shape, although more coarsely than the linear/snake structures. The idea of approximating a given shape with a linear structure is to draw a non-intersecting curve consisting of horizontal and vertical line segments. Each line segment is a linear chain of basic tiles depicted in Figure 1(a). At first glance, the wave structures seem more restricted than linear structures, as the line segments they use are very short and have the same size (3 tiles long). However, one can simulate arbitrary long line segments with wave structures forming a zig-zag pattern, cf. Figure 5.

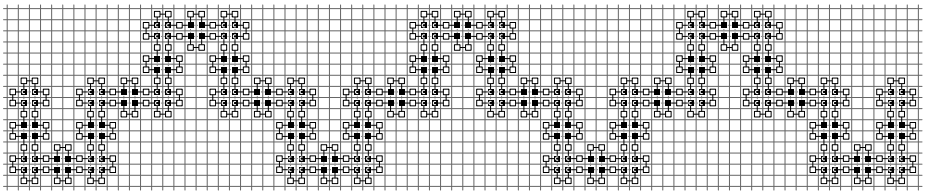

Fig. 5. Simulation of a straight line segment with a wave structure.

We prove that the proteins for the wave structures are stable in the HPC model. Our techniques to achieve this include (i) the case analysis (also used in Gupta *et al.* [1]) and (ii) the induction on diagonals. Furthermore, to increase the power of the case analysis technique, we used a program called "2DHPSolver" for semi-automatic proving of hypothesis about the folds of proteins of the designed structures developed in [2]. Note that 2DHPSolver can be used for all three models: HP, HPC and strong HPC by setting the appropriate parameters.

3 Stability of the wave structures

In this section we prove that the protein of any wave structure is stable. In the proof we will use a concept of saturated structures and 2DHPSolver tool developed in [2]. Let us briefly introduce them.

3.1 Saturated folds

The proteins used by Gupta *et al.* [1] in the HP model and the wave proteins in HPC have a special property. The energy of their native folds is the smallest

possible with respect to the numbers of hydrophobic cysteine and non-cysteine monomers contained in the proteins. We call such folds *saturated*. In saturated folds all parts of energy function produce minimum possible values. This means: (i) every hydrophobic monomer (cysteine or non-cysteine) has two contacts with other monomers; (ii) there is a sulfide bridge matching containing all or all but one cysteine monomers. Obviously, a saturated fold of a protein must be native, and furthermore, if there is a saturated fold of a protein, then all native folds of this protein must be saturated.

3.2 2DHPSolver: a semi-automatic prover

2DHPSolver is a tool for proving the uniqueness of a protein design in 2D square lattice under the HP, HPC or strong HPC models developed in [2]. 2DHPSolver is not specifically designed to analyze the wave structures or even the constructible structures. It can be used to prove the stability of any 2D HP design based on the induction on the boundaries. It starts with an initial configuration (initial field) which is given as the input to the program. In each iteration, one of the fields is replaced by all possible extensions at one point in the field specified by user. Note that in displayed fields red 1 represents a cysteine monomer, blue 1 a non-cysteine monomer and finally, uncolored 1 is hydrophobic monomer, but it is not known whether it is cysteine or not.

These extensions are one of the following type:

- extending a path (of consecutive monomers in the protein string);
- extending a 1-path (of a chain of hydrophobic monomers connected with contacts);
- coloring an uncolored H monomer.

There are 6 ways to extend a path, 3 ways to extend a one-path and 2 ways to color an uncolored H monomer. For each of these possibilities, 2DHPSolver creates a new field which is then checked to see if it violates the rules of the design. Those which do not violate the design rules will replace the original field.

However, this approach will result in producing too many fields, which makes it hard for the user to keep track of. Therefore, 2DHPSolver contains utilities to assist in automatically finding an extending sequence for a field which leads to either no valid configurations, in which case the field is automatically removed, or to only one valid configuration, in which case the field is replaced by the new more completed configuration. This process is referred to as a *self-extension*. The time required for searching for such extending sequence depends on the depth of the search, which can be specified by user through two parameters "depth" and "max-extensions". Thus, leaving the whole process of proving to 2DHPSolver by setting the parameters to high values is not practical as it could take enormous amount of time. Instead, one should set parameters to moderate values and use intuition in choosing the next extension point when 2DHPSolver is unable to automatically find self-extending sequences. Note that these parameters can be changed at any time during the use of the program by the user.

2DHPSolver is developed using C++ and its source code is freely available to all users under the GNU Public Licence (GLP). For more information on 2DHPSolver and to obtain a copy of the source codes please visit http://www.sfu.ca/ ahadjkho/2dhpsolver/.

3.3 Proof

Let S be a wave structure, p its protein and let F be an arbitrary native (i.e., saturated) fold of p.

Define a path in F as a sequence of vertices such that no vertex appears twice and any pair of consecutive vertices in the path are connected by peptide bonds. A cycle is a path whose start and end vertices are connected by a peptide bond. For $i \in \{0, 1, 2\}$, an i-vertex in the fold F is a lattice vertex (square) containing a monomer i. For instance, a square containing a cysteine monomer in F is called a 2-vertex. An H-vertex is a vertex which is either 1-vertex or 2-vertex. Define a 1-path in F to be a sequence of H-vertices such that each H-vertex appears once and any pair of consecutive ones form an HH contact. A 1-cycle in F is a 1-path whose first and last vertices form an HH contact. A 1-cycle of length 4 is called a core in F.

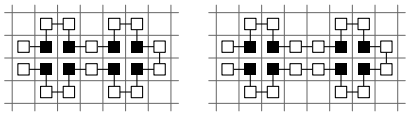

Fig. 6. Configurations with correctly aligned cores.

A core c is called *monochromatic* if all its H-vertices are either cysteines or non-cysteines. Let c_1 and c_2 be two cores in F. We say, c_1 and c_2 are adjacent if there is a path of length 2 or 3 between an H-vertex of c_1 and an H-vertex of c_2. We say c_1 and c_2 are correctly aligned if they are adjacent in one of the forms in Figure 6.

In what follows we prove that every H-vertex in F belongs to a monochromatic core and the cores are correctly aligned.

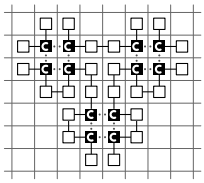

Fig. 7. Configuration with misaligned cores.

Lemma 1. *Every* H*-vertex in F belongs to a monochromatic core and either all the cores are correctly aligned or there are three cores in F that are not correctly aligned while all other cores are correctly aligned and these three cores form the configuration depicted in Figure 7.*

Proof. For any integer i, let SW_i be the set of lattice vertices $\{[x, y]; x + y = i\}$. Let m be the maximum number such that SW_i, $i < m$ does not contain any H-vertex, i.e., SW_m is a boundary of diagonal rectangle enclosing all H-vertices.

We start by proving the following claim.

Claim. Every H-vertex in F belongs to a monochromatic core.

Proof. We prove the claim by induction on SW_k, i.e., we prove that for every k and every H-vertex v on SW_k, v is in a monochromatic core. For the base case, consider smallest k such that for $i \geq k$, there is no H-vertex on SW_i. Then the claim is trivially true. For induction step, it is enough to show that for every k, if for every H-vertex v on SW_i, $i > k$, v is in a monochromatic core, then for every H-vertex w on SW_k, w is on a monochromatic core c.

Fix k and by induction hypothesis, assume that for every H-vertex v lying on SW_i, where $i > k$, v belongs to a monochromatic core. Consider an H-vertex w on SW_k. We show that if w is not on a monochromatic core then we see a subsequence in F which is not in p or an unpaired cysteine monomer. This is done by enumerative case analysis of all possible extensions of this configuration and showing that each branch will end in a configuration that has a subsequence not in p or has an unpaired cysteine monomer.

This process requires the analysis of many configurations which is very hard and time consuming to do manually. Therefore, we used 2DHPSolver to assist in analyzing the resulting configurations. The program generated proof of this step of the induction can be found on our website at
http://www.sfu.ca/ ahadjkho/2dhpsolver/core-monochromatic-proof.

Finally we showed the following claim using the 2DHPsolver tool.

Claim. Let c_1 and c_2 be two adjacent monochromatic cores in F. Then either c_1 and c_2 are aligned correctly or there is a third core c_3 such that c_1, c_2 and c_3 form the configuration in Figure 7.

The program generated proof of this claim can be found on our website at http://www.sfu.ca/ ahadjkho/2dhpsolver/core-alignment-proof.

The main result follows from the previous lemma and the proof of the main result in [2].

Theorem 1. *Every* H*-vertex in F belongs to a monochromatic core and all the cores are correctly aligned. Hence, $F = S$, i.e., all wave structures are stable.*

4 Conclusions

In this paper we introduce a robust subclass of constructible structures introduced by Gupta *et al.* [1] able to approximate any given shape, and refine these structures for the HPC model [2] and prove that these structures are stable. This result shows that use of cysteines in the design of proteins might help to improve their stability. To further verify this, in the future, we would like to extend our results to 3D lattice models and test them using existing protein folding software.

References

1. A. Gupta, J. Maňuch and L. Stacho, *Journal of Computational Biology* **12**, 1328 (2005).
2. A. R. A. Hadj Khodabakhshi, J. Maňuch and A. Gupta, Structure-approximating design of stable proteins in 2D HP model fortified by cysteine monomers, in *Proc. of APBC 2008*, 2008.
3. K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas and H. S. Chan, *Protein Science* **4**, 561 (1995).
4. K. A. Dill, *Biochemistry* **29**, 7133 (1990).
5. K. A. Dill, *Biochemistry* **24**, 1501 (1985).
6. P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni and M. Yannakakis, On the complexity of protein folding, in *Proc. of STOC'98*, 1998.
7. B. Berger and T. Leighton, *J. Comp. Biol.* **5**, 27 (1998).
8. O. Aichholzer, D. Bremner, E. Demaine, H. Meijer, V. Sacristán and M. Soss, *Computational Geometry: Theory and Applications* **25**, 139 (2003).
9. Z. Li, X. Zhang and L. Chen, *Appl. Bioinformatics* **4**, 105 (2005).
10. B. Hayes, *American Scientist* **86**, 216 (1998).
11. R. Jaenicke, *Eur. J. Biochem.* **202**, 715 (1991).
12. Y. Liou, A. Tocilj, P. Davies and Z. Jia, *Nature* **406**, 322 (2000).
13. G. C. Rodakis and F. C. Kafatos, *Proc. Natl. Acad. Sci. USA* **79**, 3551 (1982).