Indirect Space Sampling Genetic Algorithm for the HP Model for Protein Folding

By Yonatan Ben-Simhon
Department of Computer Science, Cornell University, Ithaca, NY
bensimho@cs.cornell.edu
http://www.cs.cornell.edu/~bensimho

Abstract:

Lattice models make simplifications to protein structure that form a relatively tractably searchable space in which polymers can fold in a quasi-biological manner and are thus used in protein folding and protein evolution studies[1].
Since even with the simplifications of the lattice models the problem of finding the energetically optimal conformation of a polymer was proved to be NP-complete many types of search algorithms were implemented on this problem including Monte-Carlo methods, pruning methods and a variety of evolutionary algorithms.
Existing search methods represent the structure in ways that differ in the details, but all keep the actual conformation and conduct their search on the conformations space. Since conformations must be self-avoiding difficulties in moving between acceptable structures was reported repeatedly.

I start by presenting a new approach that samples the conformation space indirectly and thus allows easier movement in the search space. This principal can be applied in many of the existing search methods. Later I introduce a series of heuristics that limit the search space and apply a simple genetic algorithm to evaluate this technique. I will compare it to a similar genetic algorithm that searches directly in the conformation space and report the results.

Future research would include adding a local search algorithm as a second stage to try and improve on the final result of the genetic algorithm. The local search approach I suggest is a chain-growth based method with a heuristic function that is highly dependant on a good starting conformation – a condition we can expect to have at the end of the first stage.

1) Introduction:

Proteins can be viewed as microscopic machines that are formed and act in the cells of living creatures. One of the basic principals of molecular biology originally suggested in the late 1800's by Emil Fischer suggests that the function of a protein is determined by its structure. Proteins are polymers – chains of amino acids that fold into a specific structure. Though the composition of a protein is held in a one dimensional chain made of an alphabet of 20 characters, its 3 dimensional structure is what gives us the information regarding its function. It happens very often that we can identify a proteins' sequence without knowing its structure. The procedure of finding a protein structure or/and function is costly and not always possible [2]. Many algorithms attempting to find a proteins structure from its sequence have been developed [3] but they are of poor quality when it comes to predicting structures of proteins without known homologues [3,4]. Many AB-Initio methods for protein folding attempt to fold the protein sequence by minimizing an energy function [5].

2) The 2D HP lattice model for protein folding:

Lattice models were first introduced by Dill [6] as a means to study the process of protein folding. While the protein chain of amino acids can freely move in space and has unlimited amount of possible conformations, lattice models allow the amino acids in the chain to be located only at discrete locations on a lattice. This way the search space is tractable since the number of possible conformations is finite. The price for this simplification is much lesser accuracy and a model that losses much of its power with direct biological meaning. A further simplification of the model is reducing it from 3 dimensions to 2 dimensions, that is from a cubic lattice to a square lattice. Another simplification is reducing the alphabet from 20 symbols representing the 20 amino acids, to 2 symbols representing the electrical charge category of the amino acid's side chain: P for polar and H for hydrophobic. The reasoning behind the last simplification is that it is believed that the major force playing in protein folding is that of the surrounding polar environment (water) repels the hydrophobic residues creating globular proteins with a hydrophobic core "hiding" from the surface.

A 2D HP lattice model is associated with an energy function that is responsible to maintain a stable protein structure. Such an energy function should utilize its ability to distinguish between only 2 types of letters to favor conformations with a hydrophobic core. For this we must introduce the concept of contacts. A contact is defined as any 2 amino acids on the lattice that are adjacent on the lattice but not directly connected on the amino acid chain. The most common such function is one that assigns an energy of –1 to all H-H contacts (contacts in which both amino acids are hydrophobic). This function favors conformations in which the hydrophobic residues are clustered together. This phenomena leads to a hydrophobic core surrounded by polar "surface" residues. Another less common energy function includes an additional polar repulsion by setting a positive energy of ½ to contacts that are not H-H. This function yields structures that are generally less compact.

For visualization purposes it is common to draw the layout of the chain on the lattice with black boxes representing hydrophobic residues and white boxes representing polar residues. Figure 1 demonstrates a layout of a chain on a lattice using the common energy function.

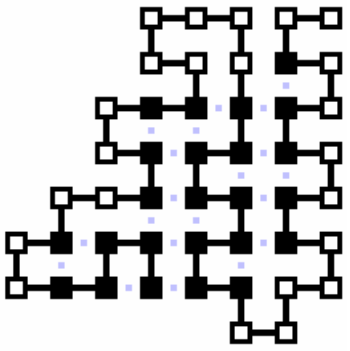


Figure 1. An optimal 2D lattice layout of the 47 amino acids long sequence PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPHHPPHHPPHPPHHHHH. The purple dots represent H-H contacts. Using the common energy function this structure yields an energy of –20. Two biological phenomena can be viewed in this conformation, a buried hydrophobic core and a helix-like secondary structure on the right hand side.

Though the 2D HP model is not representation for real life protein it got its popularity due to its simplicity and that fact that it can represent many important protein-like attributes such as secondary structure and active sites. It was also shown to demonstrate protein like behavior under energy landscape constraints[].

A protein is considered to favor the conformation in which it has the minimal energy, so for most applications we are mainly interested in finding the lowest energy conformation of a given string on a lattice. There is another interesting reverse problem of finding the optimal sequence for a given structure, but this is out of the scope of this paper.

3) Optimization Algorithms:

Finding the minimum energy conformation of a given string on a square lattice was proven to be an NP-complete problem [7]. Since the problem is formulated so elegantly and is related to biology, there were many papers written about algorithms for finding the best conformation.

Techniques for solving the 2D HP model can be generally separated into 2 groups: chain growing and conformation space sampling. While chain growing methods attempt to cover the full conformation space, sampling methods attempt to only sample promising regions of the conformation space.

Chain growing methods include full space search [6], pruned searches [13], hydrophobic zipper [11] and more [12]. Space sampling algorithms include genetic algorithms[8,9], simulated annealing[9] and more [14]. A good summery of previous and state of the art search algorithms for the 2D HP problem can be found at [8].

One of the first and most important papers written about this problem was a comparison between a Monte-Carlo algorithm (simulated annealing) and a genetic algorithm [9]. This paper by Ron Unger and John Moult shows how genetic algorithms seem to have better performance than simulated annealing on this problem. As far as I can tell they were the first to publish a genetic algorithm approach to this problem. They also introduced an approach that that is kept to date with all space sampling approaches in which the structures are represented as strings holding directional information. So instead of dealing with 2 dimensional structures they deal with a 1 dimensional string. 2 common schemas for structure representation are UPLR and FCA. The first method gives absolute directions: Up, Down, Left and Right from the current position. The first amino acid can be placed at an arbitrary point. The second method gives directions with respect to the current direction in the form of keeping Forward, turning Clockwise or Anti-clockwise. This facilitates the implementation of regular genetic algorithms since the structures are represented as character strings. It was shown [10] that the last representation is more advantageous than the first.

From the abundance of papers attempting to solve this problem via space sampling there is one clear problem that arises over and over. The conformation space as represented by these strings is disconnected. Furthermore there is an abundance of such strings that are not valid since they represent a layout that is not self-avoiding (i.e. the chain crosses over itself). This restriction on the sampling space makes the search algorithms perform badly. One example from the texts [10] shows how a sequence with 2 optimal conformations that seem close (similar) to the eye and are close in infeasible

space are in fact very distant in feasible space (Figure 2). To get from the first to the second conformation one must open the conformation, mutate it and close it up again which takes many steps.

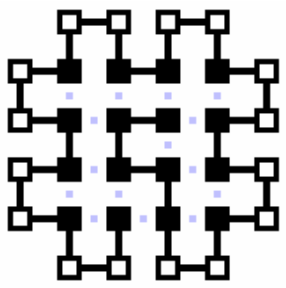
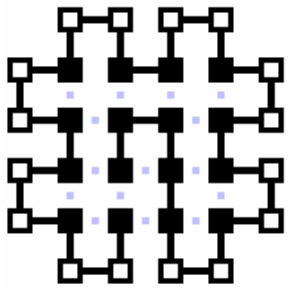
a)  b) 

Figure 2. This sequence of length 32 has 2 optimal conformations with energy -15. Though these conformations are close in infeasible space they are distant in feasible space

Most older algorithms simply did not accept non self-avoiding conformations, but a more common approach today is to allow non self avoiding conformations, but to panelize them. This allows the structures to cross conformational barriers as described above.

There is a benchmark on which these algorithms are tested that was introduced in [8]. This benchmark has no biological reasoning and was accepted simply because it was the first one to be published. A new benchmark is being created today in which sequences come from short globular proteins sequences that are "translated" into HP sequences and from sequences generated using a Markov chain with globular proteins amino acid probabilities.

Tables 1 and 2 show the benchmark sequences and a comparison of a few of the existing algorithms [8,11,12,13]:

#	length	sequence
2	20	hphpphhphpphphhpphph
3	24	hhpphpphpphpphpphpphhh
4	25	pphpphhpppphhpppphhpppphh
5	36	ppphhpphhpppppphhhhhhpphhpppphhpphp
6	48	pphpphhpphhpppppphhhhhhhhhhpppppphhpphhpphpphhhhh
7	50	hhphphphphhhhhpppphpppphpppphpppphpppphphhhhhphphphhh
8	60	pphhhphhhhhhhhhpppphhhhhhhhhhhphpppphhhhhhhhhhhhhpppphhhhhhphhphp
9	64	hhhhhhhhhhhhphphpphhpphhpphpphhpphhpphpphhpphhpphpphpphhhhhhhhhhhh

Table 1. The 2D HP benchmark

sequence	length	MC [8]		GA [8]		CI [11]		CG [12]		HIT [13]	
		time	max	time	max	time	max	time	max	time	max
2	20	?	9	5	9	0	9	2	9	0	9
3	24	?	9	6	9	0	9	7	9	0	9
4	25	?	8	4	8	0	8	5	8	0	8
5	36	?	13	54	14	6	14	132	14	1	14
6	48	?	20	?	22	35	23	378	23	7	23
7	50	?	21	3180	21	0	21	18600	21	285	21
8	60	?	33	?	34	60	35	5820	35	85	36
9	64	?	35	?	37	?	40	546	42	132	39

Table 2. Comparison of some existing algorithms. **'**max' is the maximum number of contacts of best conformation reached. 'time' is the cpu time in seconds, where known as reported in the papers.

4) Algorithmic design:

What I present in this paper as a novelty is a technique that can allow space sampling algorithms to work with better efficiency. The idea is to reduce the prevalence of invalid conformations and yield higher fertility rate in genetic algorithms. My technique is also meant to connect the search space in a much better way thus dealing with the hardest hurdles of sampling method approaches to this problem.

The intuition comes from the fact that these structures represented as direction strings are very rigid and thus tend to cross over themselves even when there is free space for the structure to expand near by. If we could find a structure representation that allows the structures to be more flexible we would avoid much of these invalid conformations. The gist of the idea is not working with conformations directly at all! Instead I suggest working on another structure – matrices. If we can find a function that easily translates from matrix space to chain conformation space that also promises us high fertility then we are good.

The function I am suggesting here is a simple follow-the-highest-value approach. If all the values in a matrix are different, start by placing the first bead of the chain on the highest valued location in the matrix. For the following beads locate the highest valued neighboring element and place the bead there. Repeat this process until the chain is exhausted or until there are no available neighboring locations. If all the amino acids were used this matrix produced an acceptable conformation. If the chain growth terminated due to absence of available neighbors than this matrix is considered invalid (similarly to a non self avoiding conformation). Since the building of the chain on the matrix is dynamic – the chain can grow into any vacant direction – this technique intuitively will have a lower percentage of invalid conformations. After a valid conformation is built on the matrix we can calculate its energy following any energy function we chose.

Figure 3 is an example of how this matrix-to-conformation function works, while figure 4 shows how to calculate an energy function from it.

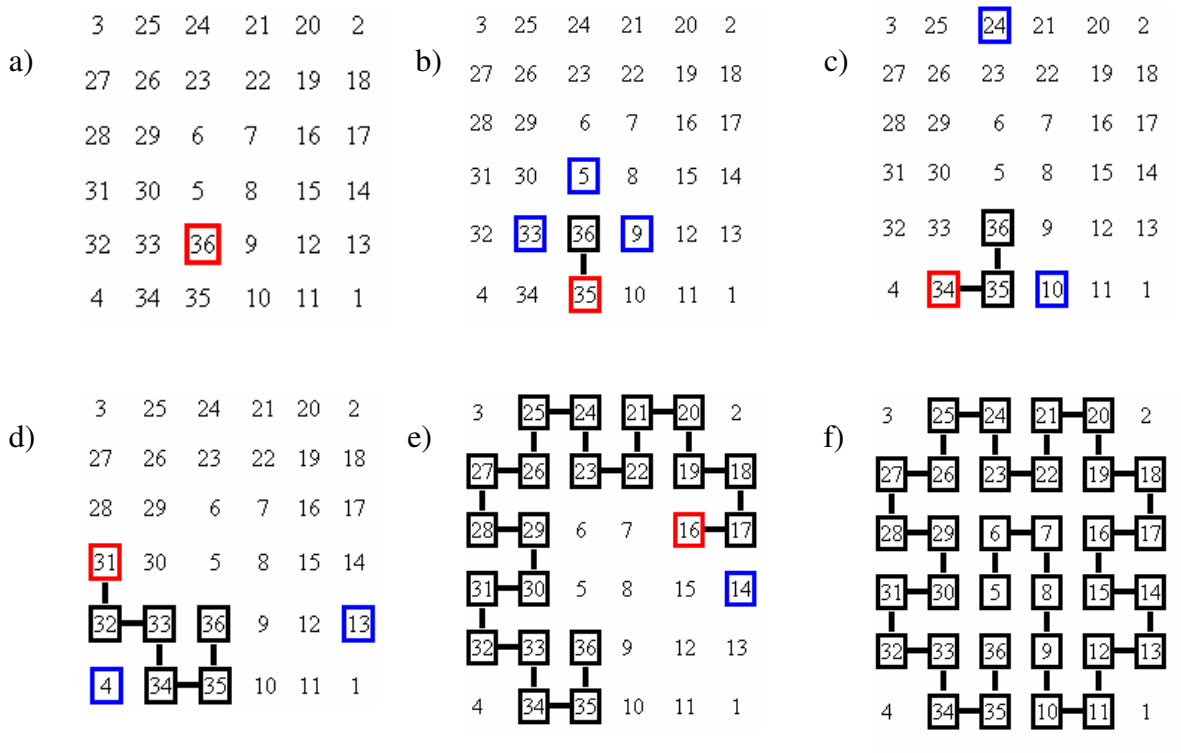


Figure 3. An example for how the matrix-to-conformation function works for a string of 32 beads on a given matrix. Black boxes are locations occupied by the chain. Red boxes indicate the highest valued neighbor – the position in which the following bead will be placed. Blue represents the other vacant neighbors of the current position of lower value than the red one. At (a) we start by locating the highest valued slot in the matrix (#36), we select it and follow to its highest valued neighbor (#35) at (b). We continue this way (c)-(e) and finally at (f) we have finished placing all 32 beads to get a valid conformation.

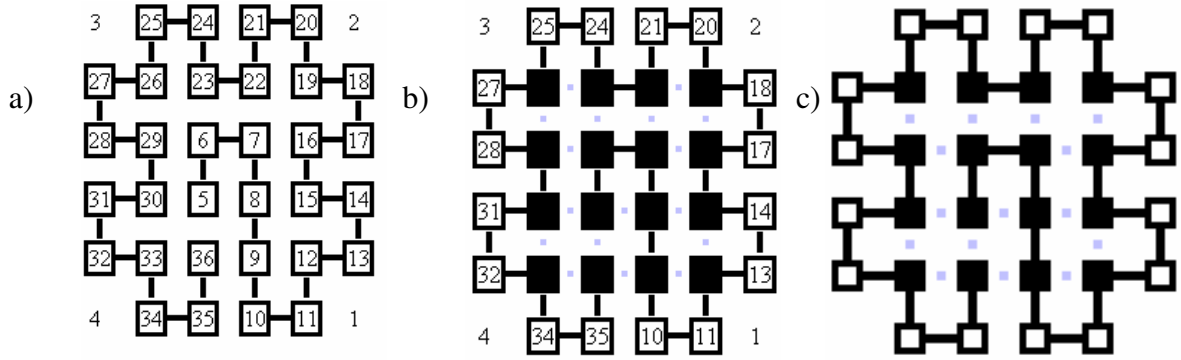


Figure 4. Calculating the energy function for a matrix. After the chain has physically been lain on the matrix to create the structure (a) the HP coloring is added according to the query string (b) and the energy is calculated by the desired energy function according to the structure (c).

There are a few important attributes that make this matrix to matrix-to-conformation that make it a good candidate. It is an onto function to the protein space, it is highly connected and it easily crosses physical conformational boundaries.

The first and most important fact is that this function covers all chain conformation space. If this was not the case then our search would be limited to the conformations in the range of the function which might not include the optimal conformation. It is easy to prove that the function is indeed onto the conformation space.

Claim: for a conformation C of length n, there is a corresponding matrix M such that f(M)=X where f is the above matrix-to-structure function.

Proof by building: For the given conformation X on a lattice for a sequence of length n, we start by creating M as a matrix of zeros. At the location of the first bead of the conformation X place the value n in the matrix M. Then at the location of the second bean place the value n-1 on the matrix. Continue in this fashion placing the value n-i+1 on the matrix M at the location of the i^{th} bead in conformation X.

When we use the matrix-to-conformation function we start at the highest value location. By the way we built the matrix M we are know that the highest value on it is n, and that it is located at the same location of the first bead of X. Of the neighbors of this location on M we know by the building that there is a neighbor with the value n-1 at the location of the second bead of X. We also know that this is the heist valued unoccupied position in M since the value n is occupied and the rest of the values in M are 0 through n-1 by the way we built M. So we place the second bead in that location. Inductively, after placing bead i on M at a location with value n-i+1 at the corresponding location to Xi, we know by the built that it has a neighbor of value n-i. We also know that n-i is the highest valued unoccupied location in M since all the values n to n-i+1 are already occupied, so of the unoccupied neighbors of bead i this one is the highest valued unoccupied location. Thus we continue and place bead i+1 there at the location valued n-(i+1)+1=n-i. We end at placing the n^{th} (last) bead on the location with value n-n+1=1 on M. See figure 5 for an example.

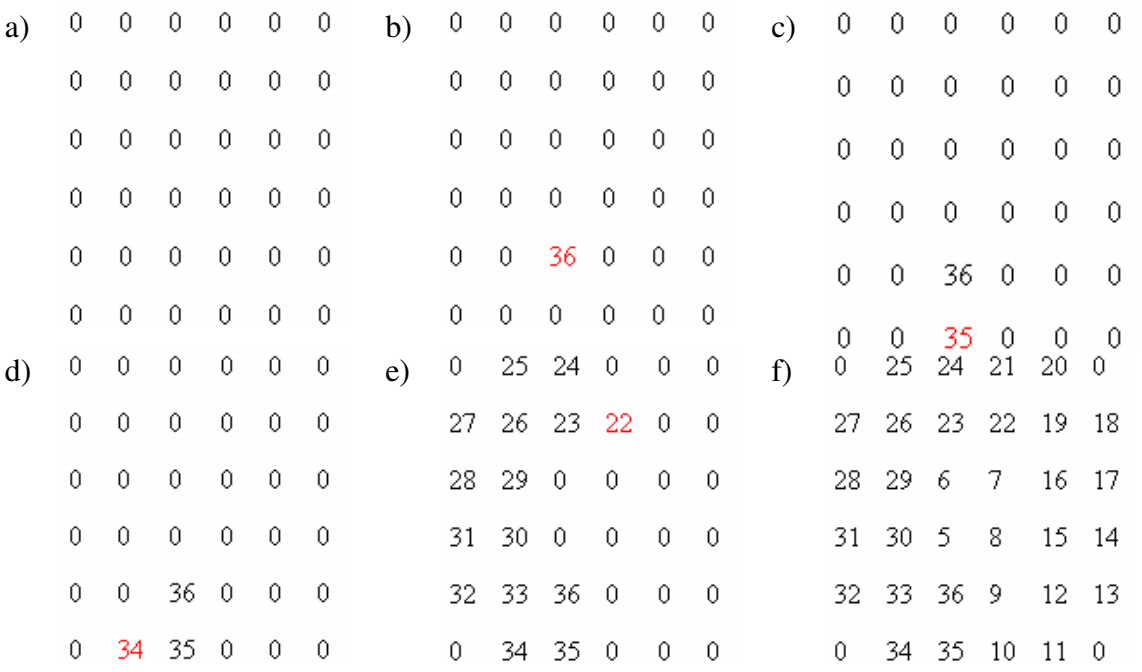


Figure 5. An example of the proof for the claim that every conformation has a corresponding matrix. Here our goal is to create the conformation from Figure 2a. We start with a matrix of zeros M (a). Place the highest value (36) in the location of the beginning of the conformation (b). Put a value that's smaller by one (35) at the location of the next bead (c). Continue inserting decreasing values along the path of the conformation (d,e) until you reach the final matrix (f) which yields the structure we are looking for (see in figure 4 that this matrix indeed corresponds to this structure)

So we showed that every conformation has at least 1 Matrix that yields it but clearly this is not the only matrix that corresponds to the given conformation. Take the matrix M we built above and add ½ to all of its cells and you will get a different matrix that yields the exact same conformation. In fact there is an infinite amount of matrices that correspond to every conformation. An open question is whether all conformations have equal likelihood to be generated by (random) matrices or is there a bias? My intuition says that there has to be a bias since of some sort since the conformational representation is unbiased yet the matrix representation favors against non self-avoiding conformations. So conformations that are distant from non self-avoiding conformation (e.g. a strait line) will be favored against in the matrix representation.

Since this matrix representation is not bound to conformational constraints it crosses physical conformational boundaries much easier than direct conformational representations. The fact that the matrix-to-conformation function is not rigid but a fluidly grows the chain on it allows it to bypass local obstacles. If we look again at the problematic conformations as given in [10] (Figure 2), then with matrix representation a single mutation moves us between these 2 structures. Figure 6 shows this example.

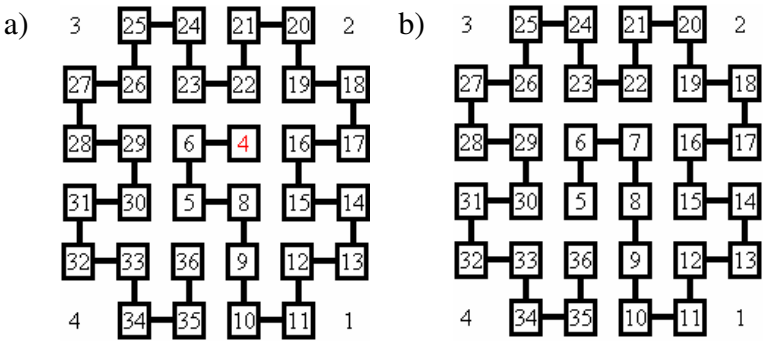


Figure 6. Here we see how we can move between the structures presented in Figure 2 using only a single mutation (marked in red) in the matrix representation. These conformations are closer to each other in matrix representation than in conformation space even when non self-avoiding conformations are allowed.

Though not all conformations are 1 mutation distance apart in matrix representation, intuitively it seems that similar looking conformations are close in matrix representation space. If instead of looking at single point mutations we allow ourselves to look at conformation connectivity through crossovers we can see that the conformation space is fully connected under crossovers. If we use a crossover method that takes a weighted average of 2 matrices as the crossover product of the matrices, for any 2 given M_1 and M_2 we can build a matrix M_3 such that the crossover of M_1 and M_2 will yield M_3 by using the formula $M_3 = 2(M_2 - ½ M_1)$. Proof by plugging this M_3 into the formula for the crossover: $M_2 = ½ (M_1 + M_3) = ½ (M_1 + 2(M_2 - ½ M_1)) = ½ (M_1 + 2(M_2 - ½ M_1)) = ½ (M_1 + 2M_2 - M_1) = ½ (2M_2) = M_2$.

This method of sampling on the matrix space instead of conformations does have drawbacks. Since prior to creating the matrix we can not say to which direction the conformation will grow, we might need an n-by-n matrix to represent a conformation for a sequence of length n. This can be seen in the extreme case of fully extended chains which must have a length of n, since it might be extended horizontally or vertically the matrix must be of size n-by-n to be able to accommodate all conformations. This means we have to use representations of size n^2 which consume greater memory and take greater time to mate when using genetic algorithms. In cases where the size of the population is similar to the length of the sequence this slows the algorithm considerably.

An even more alarming problem is that using this representation we are trying to optimize a structure of size n^2 instead n, which means we have many more parameters to optimize, our actual search space increased dramatically and we can expect a need for many more generations of any space sampling algorithm in order to converge.
We will deal with these drawbacks in the next part.

5) Heuristics:

We introduce a heuristic to improve the running time and reduce the search space using this matrix representation. Based on the notion that the 2D HP model simulates globular proteins behavior in which a hydrophobic core is built, we can stipulate that the globular protein like optimal conformations will be relatively compact. When looking at optimal structures for sequences from the literature such as [3,8,9,10,13] and more we

can see that optimal conformations are indeed compact with a width to length ration of no more than 2 to 1. That means that all these structures fall within a rectangle whose long edge is no more than twice the length of its short edge.

For a sequence of length n to fall into a rectangle with this ratio, with one edge being of length A and the other 2A, we get that $n \leq A*2A=2A^2$, so for $2A=\sqrt{2n}$ we get $2A^2 = n$. This means that if all optimal structures for sequences of length n fall within a rectangle with 2 to 1 ratio, then they fall into a rectangle with the long edge of size $\sqrt{2n}$. We can now claim that any optimal conformation under this assumption will have a $\sqrt{2n}$ by $\sqrt{2n}$ matrix that represents it. Such a matrix is of size of $\sqrt{2n} * \sqrt{2n} = 2n << n^2$ and we are back to dealing with linear length representation.

But even if we accept that all optimal confirmations adhere to this assumption this heuristic has a drawback. When we reduce the matrix size there is less area for the chain to grow on it which will increase the relative number of matrices yielding invalid (non self-avoiding) conformations.

Another heuristic we introduce to limit the search space specifically for genetic algorithms is normalizing the matrices such that the sum of all the elements in a matrix is 1 and all the values are positive. This reduces the full space connectivity mentioned before, but practically has very good behavior and promises that matrices won't dominate one the other when using weighted average crossovers.

6) Results:

To adequately compare the conformation representation space and the matrix representation space we used both of the representations in a genetic algorithm. The genetic algorithm we implemented was that reported in [3]. A population of size 200 is randomly generated. Parents are selected for mating with the probability of parent i with conformation k_i yielding energy E_i to be chosen being $P_i = E_i / \Sigma_{j=1..n} E_j$. At each generation 20 mutations are performed on each member. The probability to accept a mutation or a crossover is $\exp[(E_{parents\_avg} - E_{child})/T_k]$ where T_k is the temperature at generation k. Initial temperature is set on 2 with a decreasing multiplication factor of 0.99 every generation.

Of the many techniques to implement crossovers in matrix representation, we chose to take the average of the 2 matrices as the offspring (evenly weighted average). As mutation we implemented it by exchanging locations of 2 values in the matrix. For the conformation representation we chose the FCA representation that was reported at [10] as having greater success (even though the original algorithm as presented by Moult & Unger used the UDLR representation).

When comparing the algorithms instead of measuring the run time in cpu seconds we report only the amount of invalid conformations reached. Since both representations are used in the same algorithm, with the same sized population and the same number of generations the real time performance difference will be visible through the number of invalid conformations reached. Since both representations are run with the same algorithm the energy of the final conformation reached is a good performance evaluator for the representation.

We ran the algorithm using the 2 representations on the sequences from the benchmark (see table 1). When using the matrix representation we played with the size of

the matrix as a parameter. Parameter F represents the size factor of the matrix with respect to the sequence length. As noted in section 5 describing the matrix size heuristic the matrix has to be of at least size 2n. Larger matrices will have less non self-avoiding instances but will have a larger space to optimize over.

The results of running the algorithm with the different representations are presented in table 3. It is clear that the matrix representation outperforms the conformation representation. For F=3 the number of invalid conformations reached (that is also equivalent to the running time) is about one size of magnitude smaller than that of the conformation representation, while the energy of the optimal conformation is always superior when using the matrix representation.

Alg / Seq	Matrix			Conformations
	F=2	F=3	F=4	
Seq 6 (48)	17	17	16	14
	1.26M *	520K	304K	5.3M
Seq 7 (50)	16	15	14	15
	754K *	246K	229	5.7M
Seq 8 (60)	28	28	25	23
	1.50M *	686K	401K	9.4M
Seq 9 (64)	27	26	24	20
	2.00M *	688K	298K	8.6M

Table 3. Comparison of performance of the algorithm with the 2 representation methods. The matrix representation was tested with size factors F equal 2, 3 and 4.

It is important to note that the algorithm did not find the optimal configuration for any of the sequences with any of these sequences. This is due to the algorithm alone. In future research it will be interesting trying to use this representation with a better performing genetic algorithm or simulated annealing algorithm.

7) Discussion:

These results prove that the new representation is advantageous compared to the existing conformation representation technique.
There is still much room to experiment with the matrix size factor F. There is also room to experiment with more crossover techniques such as physical area switching between matrices. As mentioned before it should also be interesting to use this representation with one of the current state of the art algorithms and see if it improves performance there too. It is also easy to implement on the 3D cubic lattice HP model and with other energy functions.

I plan on adding a local search algorithm that uses the conformation found at the end of this algorithm as a starting point and using a heuristic for pruning the search tree to quickly traverse through the conformation search tree. The principles of this local search are presented at [12].

Refrences:

[1] Blackburne BP, Hirsta JD, “Population dynamics simulations of functional model proteins” J. Chem. Phys. 123 (2005), 154907

[2] Backer D, Sali A, “Protein Structure Prediction and Structural Genomics” Science 294 (2001), pp. 93-96

[3] Moult J, Fidelis K, Zemla A, Hubbard T, “Critical assessment of methods of protein structure prediction (CASP)-round V” Proteins 53 (2003), pp. 334-339

[4] Lemer CM, Rooman MJ, Wodak SJ, “Protein structure prediction by threading methods: evaluation of current techniques” Proteins 23(1995), pp. 337-355

[5] Hardin C, Pogorelov TV, Luthey-Schulten Z, “Ab initio protein structure prediction” Curr. Opin. Struct. Biol. 12 (2002), pp. 176–181

[6] Dill KA, “Dominant Forces in Protein Folding” Biochemestry 29 (1990), pp. 7133-7155.

[7] Crescenzi P, Goldman D, Papadimitriou C, Piccolboni A, Yannakakis M, “On the Complexity of Protein Folding” J. Comp. Bio. 5 (1998), pp. 409-422

[8] Shmygelska A, Hoos HH, “An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem” Bioinformatics 30 (2005)

[9] Unger R, Moult J, “Genetic algorithms for protein folding simulations” J. Mol. Biol. 231 (1993), pp. 75- 81.

[10] Krasnogor N, Hart WE, Smith J, Pelta DA, “Protein structure prediction with evolutionary algorithms”. Proc of the Genetic and Evolutionary Computation conference 1999, pp.1596-1601.

[11] Beutler T, Dill KA “A fast conformational search strategy for finding low energy structures of model proteins” Protein Science (1996) 5, pp. 2037- 2043

[12] Toma L, Toma S. “Contact interactions method: A new algorithm for protein folding simulations”, Protein Science (1996)5, pp. 147- 153.

[13] Ben-Simhon Y, “A pruned search tree for approximating a solution of the 2 dimensional HP model for protein folding”, http://www.cs.cornell.edu/~bensimho/lattice

[14] Bastolla U, Fravenkron H, Gestner E, Grassberger P, Nadler W, “Testing a New Monte Carlo algorithm for the protein folding problem”, Proteins 32 (1998) pp. 52-66