Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Mixed Heuristic Local Search for Protein Structure Prediction

Swakkhar Shatabda and **M. A. Hakim Newton** and **Abdul Sattar**
Queensland Research Lab, National ICT Australia
Institute for Integrated and Intelligent Systems, Griffith University
{*s.shatabda,hakim.newton,a.sattar*}*@griffith.edu.au*

Abstract

Protein structure prediction is an unsolved problem in computational biology. One great difficulty is due to the unknown factors in the actual energy function. Moreover, the energy models available are often not very informative particularly when spatially similar structures are compared during search. We introduce several novel heuristics to augment the energy model and present a new local search algorithm that exploits these heuristics in a mixed fashion. Although the heuristics individually are weaker in performance than the energy function, their combination interestingly produces stronger results. For standard benchmark proteins on the face centered cubic lattice and a realistic 20×20 energy model, we obtain structures with significantly lower energy than those obtained by the state-of-the-art algorithms. We also report results for these proteins using the same energy model on the cubic lattice.

Introduction

Proteins fold into three dimensional (3D) structures to perform various functions. Mis-folded proteins cause many critical diseases such as Alzheimers disease, Cystic fibrosis, and Mad Cow disease. Protein structures are of paramount importance in drug design. As per the famous thermodynamic hypothesis (Anfinsen 1973), every protein has a unique and stable 3D native structure that exhibits the minimum possible free energy. However, not much is known about the native structure or the underlying folding process. *In vitro* laboratory methods such as X-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy are normally used in determining a protein's structure. These methods are expensive, slow and failure-prone. On the contrary, native structures are known only for a small portion of the huge number of already-sequenced proteins. With the advent of recent fast computers, researchers (Dotu et al. 2011; Palù et al. 2011; Torres et al. 2007) from computational fields have, therefore, been attracted to this problem.

Computational methods often depend on known templates (*homology modeling*) or protein folds (*protein threading*). When such matching templates or folds cannot be found, the alternative is to rely only on the given primary amino-acid sequence. *Ab initio* methods take an amino acid sequence and then perform a search on the space of potential structures (called conformations) using the energy function as a search guidance. However, the search space of an *ab initio* method comprises an astronomically large number of conformations. Moreover, the actual energy function has many unknown contributing factors. As a result, there has been a growing interest in sampling a large number of decoys or candidate backbone structures with the help of a discrete lattice model and an energy model as realistic as possible. The candidate structures could be refined later by adding side chains to achieve the real structures; see (Rotkiewicz and Skolnick 2008) for further details. However, the conformational search space in this hierarchical approach still remains huge.

One great difficulty with the conformational search lies in that the energy models being used are often not very informative particularly for effective comparison of spatially similar structures during search. There exist a very simple 2×2 to a realistic 20×20 energy model and a few heuristic functions. However, search methods guided by only one heuristic function frequently get trapped in local minima and fail to find very low energy conformations. To address this, we propose to use a number of domain specific heuristic functions in a mixed fashion alongside the given energy model. One heuristic function could be useful when another is not. Moreover, switching heuristics prevents premature convergence of the search w.r.t a given heuristic. Alternating heuristic functions during search has been found useful in other domains such as planning and scheduling. In this paper, we propose several novel heuristics and show their collective effectiveness within a local search algorithm. Our results are also interesting because the heuristics that we use are individually weaker in performance than the energy function, but collectively produce stronger results.

We use a realistic 20×20 energy model described in (Berrera, Molinari, and Fogolari 2003). We separately consider 3D cubic and face centered cubic (FCC) lattices and place the α-carbon atoms on the lattice points. We develop a local search algorithm that uses two different neighborhood operators, maintains taboo over recently used amino-acid positions, and is guided by a mix of heuristics. On standard benchmark proteins and FCC lattice, our algorithm obtains structures with significantly lower energy levels compared to

the current state-of-the-art results. We also present results on large proteins (length > 90) while the best known cp-based algorithm (Ullah and Steinhöfel 2010) for the same models cannot find any structures within a reasonable time limit. On the same benchmark proteins and the same energy model, we also report results for cubic lattice for the first time in the literature. This is because existing energy models were designed actually by considering cubic lattice and we want to investigate the generality of our approach over different lattice models.

The rest of the paper is organized as follows: first we describe the problem model. Then we present related work followed by a detail description of the heuristics and our approach. Then we report experimental results and analyses. Finally, we conclude the paper outlining future work.

The Problem Model

Proteins are polymers of 20 amino acid monomers. We represent each amino acid by its α-C and assume the bond lengths to be equal to 3.8Å. We then place each amino acid on a lattice point. A lattice $\mathbb{L}$ is a set of points in $\mathbb{Z}^n$ where the points are integral linear combinations of given N basis vectors. N is called the *neighborhood count* or *coordination number of the lattice*. Two lattice points $p, q \in \mathbb{L}$ are said to be in contact or $neighbors$ of each other, if $q = p+\vec{v_i}$ for some vector $\vec{v_i}$ in the basis of $\mathbb{L}$. There exist a number of lattice models (see Fig. 1). FCC lattice is preferred to cubic lattice since the former provides higher degree of freedom for placing an amino acid and has a higher packing density (Cipra 1998). The points on FCC lattice are generated by 12 basis vectors: $\vec{v_1} = (1,1,0)$, $\vec{v_2} = (-1,-1,0)$, $\vec{v_3} = (-1,1,0)$, $\vec{v_4} = (1,-1,0)$, $\vec{v_5} = (0,1,1)$, $\vec{v_6} = (0,1,-1)$, $\vec{v_7} = (0,-1,1)$, $\vec{v_8} = (0,-1,-1)$, $\vec{v_9} = (1,0,1)$, $\vec{v_{10}} = (-1,0,1)$, $\vec{v_{11}} = (1,0,-1)$, $\vec{v_{12}} = (-1,0,-1)$ and the points on a cubic lattice by 6 basis vectors: $\vec{v_1} = (1,0,0)$, $\vec{v_2} = (-1,0,0)$, $\vec{v_3} = (0,1,0)$, $\vec{v_4} = (0,-1,0)$, $\vec{v_5} = (0,0,1)$, $\vec{v_6} = (0,0,-1)$.

Formally, for each i-th amino acid, s_i in a sequence S, the position of the amino acid takes a value, $(x_i, y_i, z_i) \epsilon \mathbb{Z}^3$. Every two consecutive amino acid monomers in the sequence are in contact, i.e. $\mathsf{contact}(s_i, s_{i+1}) = 1$, (called the *chain constraint*) and two monomers can not occupy the same lattice point (called the *self-avoiding constraint*). Thus the structure of a protein is essentially a self-avoiding walk on the given lattice.

For any given protein sequence s, the free energy of a structure c is calculated by the following equation:

$$E(c) = \sum_{j \geq i+1} \mathsf{contact}(s_i, s_j).\mathsf{energy}(s_i, s_j) \qquad (1)$$

where $\mathsf{energy}(s_i, s_j)$ is the empirical energy value between two amino-acids s_i and s_j and $\mathsf{contact}(s_i, s_j)$ is 1 when s_i and s_j are in contact, 0 otherwise. The $\mathsf{energy}(s_i, s_j)$ values are obtained from the energy matrix in (Berrera, Molinari, and Fogolari 2003). Given this model, the protein structure problem can be defined as follows: given a sequence S of length n, find a self-avoiding walk $p_1 \cdots p_n$ on the lattice that minimizes the energy defined by (1).

(a) (b)

Figure 1: Different 3D lattices: (a) cubic and (b) fcc

Related Work

Reduced models such as Hydrophobic-Polar (HP) energy model (Lau and Dill 1989) and discrete lattices have been studied extensively in the literature. Recent state-of-the-art results for these models have been achieved by using constraint programming (Dotu et al. 2011), genetic algorithms (Rashid et al. 2012), and memory-based local search approaches (Shatabda et al. 2013).

More realistic energy functions were derived by using statistical methods on the structures obtained by X-ray crystallography and NMR experiments. Miyazawa and Jernigan (1985) introduced two contact potential matrices that take into consideration all 20×20 amino-acid interactions. Later, another empirical energy matrix was proposed in (Berrera, Molinari, and Fogolari 2003). These models have been extensively used in *ab initio* methods and protein fold recognition. Using secondary information and constraint programming techniques, Dal Palu *et al.* (2004) developed a method to predict tertiary structures of real proteins. They also proposed several generalized and problem specific heuristics (Palù, Dovier, and Pontelli 2005). Later on, they addressed the problems with finite domain techniques of the existing CLP(FD) libraries and developed a highly optimized constraint solver named COLA (Palù, Dovier, and Pontelli 2007) to solve the problem with discrete lattices.

Large neighborhood search techniques (Ullah and Steinhöfel 2010) were used on top of the COLA constraint-programming solver to produce state-of-the-art results for real benchmark proteins of length < 75. In a recent work, a fragment assembly method was exploited to produce low energy structures using the empirical energy potentials (Palù et al. 2011). Among other approaches, population based methods (Kapsokalivas et al. 2009) and genetic algorithms (Torres et al. 2007) have been successfully applied to find low energy structures using empirical energy functions.

Our Approach

We developed a new local search algorithm. Local search methods are generally guided by one objective function. In our algorithm, we use a mix of different heuristic functions. Our algorithm also depends on a constrained initialization and two sets of neighborhood operators.

Heuristics

For convenience, we first define two groups of amino acid pairs. The first group consists of *affine* pairs. A contact between the amino acids of an affine pair is desirable for minimizing the free energy. The second group contains *repellent* pairs. For free energy minimization, a contact between

amino acids of such a pair is not desirable. The grouping is determined by setting a threshold energy level e_{th}. An amino acid pair (s_i, s_j) falls into the *affine* group if $\mathsf{energy}(s_i, s_j) < e_{th}$, otherwise it is a *repellent* pair. The value of $\mathsf{energy}(s_i, s_j)$ is taken from the energy matrix in (Berrera, Molinari, and Fogolari 2003). Ideally, e_{th} should be 0. However, we set it to be -0.5; this is to prevent amino acid pairs of weak affinity from coming close to each other and forming a premature compact core. Conceptually, this grouping is similar to the hydrophobic and polar groupings based on the polarity of the amino acids. However, the differences lie in that the $\mathsf{energy}(s_i, s_j)$ values are based on the potentials derived statistically from protein data banks.

1. **Minimize the sum of Cartesian distance of affine pairs:** This heuristic function is defined as:

   $$h_1 = \sum_{j>i+1:(s_i,s_j)\in\mathbb{A}} \mathsf{distance}(s_i, s_j)$$

   where, $\mathbb{A}$ is the set of affine pairs of amino acids and $\mathsf{distance}(s_i, s_j)$ is the Cartesian distance between s_i and s_j on the lattice. While using this heuristic, we select a move that minimizes the value of h_1. This heuristic thus helps move the affine pairs close to each other. This heuristic is similar to an existing heuristic in (Dotu et al. 2011) that minimizes sum of square distances between all H-H pairs in HP model. However, we do not use square distance as it hugely favors moving amino acids at the outer-side than those at the inner-side.

2. **Maximize the sum of Cartesian distance of repellent pairs:** This heuristic function is defined as:

   $$h_2 = \sum_{j>i+1:(s_i,s_j)\in\mathbb{R}} \mathsf{distance}(s_i, s_j)$$

   where, $\mathbb{R}$ is the set of repellent pairs of amino acids. While using this heuristic, we select a move that maximizes the value of h_2. This heuristic helps move the repellent pairs away from each other and thus create space for the affine pairs to fill in.

3. **Maximize the total number of contacts:** This heuristic computes the number of contacts between any pairs of amino acid.

   $$h_3 = \sum_{j>i+1} \mathsf{contact}(s_i, s_j)$$

   For this heuristic, we select a move that maximizes the value of h_3. This heuristics is intended to increase the contacts between the amino acid pairs regardless of their affinity grouping and hence increases interaction.

4. **Maximize the sum of sequence distance of contacts:** This heuristic is defined as:

   $$h_4 = \sum_{j>i+1} \mathsf{seqdist}(s_i, s_j).\mathsf{contact}(s_i, s_j)$$

   where $\mathsf{seqdist}(s_i, s_j)$ denotes the sequence distance $(j-i)$ between s_i and s_j. For this heuristic, we select a move that maximizes h_4. This heuristic is intended to bring amino acids that are apart in the sequence closer. This reduces the chance of forming contacts between amino acids that are close on the sequence (*chain neighbors*); such contacts leave the amino acids at the ends being dangling.

There are other heuristics that work well with the HP energy model. These heuristics include minimizing the distance of the amino acids from the centroid $(\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i, \frac{1}{n}\sum_{i=1}^{n} z_i)$ of the conformation; and minimizing the distance $\sum_{i=1}^{n}(x_i^2 + y_i^2 + z_i^2)$ from the origin. These two heuristics help create very compact structures; which is essential for HP model. However, after an initial study, we found that these are not suitable for the 20×20 energy model that we use in this paper.

Neighborhood Operators

Our search algorithm uses two types of operators to generate neighbors: *jump moves* and *pull moves*. Pull moves (Lesh, Mitzenmacher, and Whitesides 2003) provide a complete move-set for any type of discrete lattices. Pull moves allow the amino acid monomers of a structure to be pulled along the length. Although the pulling effect is minimized locally, this affects the interactions among a large number of amino acid pairs. Thus both the structure and its energy value could be significantly changed. Pull moves, though effective, require comparatively long time to find structures with small local changes.

One-monomer *kink jumps* or two-monomer *crank shaft* moves (Landau and Binder 2005) can quickly change the conformation. We extend these moves to design a generic *jump* move that works effectively for both single and multi-point changes and is able to re-optimize the sub-structures.

Figure 2: Contiguous (left) and segmented (right) moves

A jump move moves m amino acids to their neighboring free points on the lattice so that both the *self-avoiding* and *chain* constraints are satisfied. Jump moves can be of two types: contiguous or segmented. In a contiguous jump move, all the points involved in the move are contiguous amino acids in the sequence and in a segmented jump move the points are from several parts of the protein sequence (see Fig. 2). For $m = 1$, the jump move is identical to *kink jump* and for $m = 2$, a contiguous jump move is identical to a *crank shaft* move.

The Search Algorithm

Pseudo-code for our implementation is given in Algorithm 1. The algorithm starts with a bounded initialization. In each iteration, a move type is selected randomly and all feasible moves are generated for that move type. Pull moves are given less probability than jump moves. Stagnation occurs if there is no improvement in the global best solution for a number of iterations. This number is determined by a stagnation parameter. At each improvement in the global best solution, the stagnation parameter is set to the initial value, whereas it is multiplied by a factor at stagnation. The size of a jump move also depends on the stagnation parameter. Every time the search faces stagnation, the size of a jump move is increased and is set to 1 whenever the search makes improvement in finding a new global best. The search thus

Algorithm 1: localSearch()

```
1  initializeConformation()
2  while iteration ≤maxIt do
3      selectMoveTypeRandomly()
4      generatePotentialMoves()
5      selectCandidateHeuristic()
6      simulatePotentialMoves()
7      selectBestPotentialMove()
8      executeSelectedMove()
9      updateTabuList()
10     if stagnation then
11         jumpSize++
12     if improving then
13         jumpSize⟵ 1
```

exhibits an adaptive nature. The type of a jump move (segmented or contiguous) is selected randomly.

After the move selection, the points are selected randomly depending on the tabu list. At each iteration, a tabu list is maintained on the recently used points. After the instantiated moves are generated at each selected point, simulation is done using the selected heuristic function for that iteration. Simulation of a move temporarily calculates the changes in the heuristic functions without committing the move. Heuristic selection is made randomly in each iteration to prevent premature convergence w.r.t. a particular heuristic. A controlled and more informed switching could be useful and is considered to be a future work for the time being. Nevertheless, the structure with the best evaluation w.r.t. to the selected heuristic is then chosen for execution (i.e. committing the change) and the search proceeds (even if the resultant structure is worse than the current one). Note that the changes in the energy value do not affect the choices made during search. As noted before, the jump size changes with stagnation or improvement in the search.

Initialization

We initialize the solution by generating a valid structure that is bounded by a sphere of radius $r = \sqrt[3]{l * N/4}$, where l is the length of the protein sequence and N is the co-ordination number of the lattice. This formula is roughly derived as follows: a sphere with radius r has a volume $4/3\pi r^3 \approx 4r^3$ when $\pi \approx 3$; each unit volume roughly contains N lattice points and we want to put l points on the lattice. We start assigning valid basis vectors to the contacts between consecutive amino acid monomers and backtrack whenever there is a violation in any of the constraints.

Implementation

We implemented our algorithm in C++ using invariants provided by Kangaroo, which is a constraint based local search (CBLS) system (Newton et al. 2011). Invariants are special constructs that are defined by using mathematical operators over the variables. We use invariants to model the constraints, energy equations, and the heuristic functions. The simulation of moves, then execution, and related calculations are performed incrementally by Kangaroo. This helped us explore search strategies without significant concern about their efficient implementations.

Experiments

We ran our experiments on a cluster of computers. Each node in the cluster is equipped with Intel Xeon CPU X5650 processors @2.67GHz, QDR 4 x InfiniBand Interconnect. We compared the performance of our approach with the hybrid algorithm proposed in (Ullah and Steinhöfel 2010).In the hybrid algorithm (Ullah and Steinhöfel 2010), a random structured initialization procedure is followed by large neighborhood search using COLA solver (Palù, Dovier, and Pontelli 2007). Randomly selected large chunks of amino-acid points are selected for large neighborhood search using constraint programming and exhaustive generation. The local search follows a simulated annealing (Steinhöfel, Skaliotis, and Albrecht 2007) optimization method. For each of the benchmark protein sequences, we ran each algorithm 50 times with a fixed time limit of one hour. The stagnation parameter and jump size was initially set to 1000 and 1. The multiplying factor was kept 1.2.

Results and Analysis

To evaluate our algorithm, we have used two sets of benchmark proteins. The first set of proteins were also used in (Ullah and Steinhöfel 2010). These proteins are of length in the range 54-74. We report the best and average energy levels achieved by both algorithms in the upper part of Table 1. We also have used a second set of five proteins taken randomly from CASP9.[1] Results for these proteins are reported in the lower part of the table. These proteins are in the length range 90-160. Notice that the larger the sequence number in the table, the longer the protein sequence. For both sets of benchmark protein, we report results after running our algorithm on both cubic and FCC lattices. Note that our results are to be compared with that obtained by other *ab initio* algorithms on the same energy models and lattice models.

Results of our algorithm that uses mixed heuristics are reported in the columns titled 'heuristics'. Since the hybrid algorithm in (Ullah and Steinhöfel 2010) supports only FCC lattice, we report the results obtained by the hybrid algorithm under the column 'hybrid'. The blank values in the table indicates that no valid structure was found for the protein by the particular algorithm within the given time limit. Under the column 'e_{matrix}', we also report the energy values achieved by our algorithm when the energy function itself is used as the heuristic function to guide the search and no other heuristic is used. PDB ids and the corresponding lengths of the proteins are also shown in Table 1.

Table 1 also shows the comparatively lower energy levels in bold faced fonts. For all the proteins, the heuristics approach produces lower energy structures. Due to the expensiveness of the constraint programming methods, the hybrid method is not able to report any valid structures within the time limit for any of the proteins in the second set. We have performed statistical *t*-test with a confidence level of 95% to verify the significance of the results. For the rest of

[1] http://predictioncenter.org/casp9/targetlist.cgi

Table 1: Results for Cubic and FCC lattice for 12 proteins

seq	pdb		Cubic Lattice						FCC Lattice								
			e_{matrix}			heuristics			e_{matrix}			heuristics			hybrid		
no	id	len	best	avg	rmsd	best	avg	rmsd	best	avg	rmsd	best	avg	rmsd	best	avg	rmsd
1	4rxn	54	-63.15	-59.96	6.41	**-65.45**	**-62.52**	6.36	-153.344	-146.99	6.305	**-165.21**	**-156.32**	**6.29**	-157.70	-140.13	9.99
2	1enh	54	-55.63	-53.98	6.93	**-58.63**	**-56.41**	6.71	-145.48	-139.96	7.817	**-158.75**	**-146.69**	**6.61**	-154.24	-141.99	10.04
3	4pti	58	-79.14	-75.33	8.42	**-81.52**	**-77.19**	7.92	-198.279	-184.10	8.131	**-219.52**	**-198.42**	**7.07**	-213.70	-196.23	11.92
4	2igd	61	-74.01	-66.01	9.38	**-71.04**	**-68.96**	9.37	-172.805	-168.82	9.12	**-187.20**	**-174.19**	**9.33**	-184.29	-157.20	13.30
5	1ypa	64	-96.64	-93.24	8.41	**-100.53**	**-96.18**	7.98	-243.765	-235.75	7.42	**-249.90**	**-239.98**	**7.53**	-221.11	-208.10	13.42
6	1r69	69	-79.10	-74.86	8.54	**-82.66**	**-78.43**	6.67	-203.3	-196.83	8.27	**-213.04**	**-204.17**	**6.47**	-180.62	-165.11	14.78
7	1ctf	74	-84.15	-81.34	8.58	**-87.88**	**-84.77**	7.89	-209.518	-197.86	8.24	**-224.29**	**-213.81**	**7.23**	-204.88	-195.23	12.65
8	3mx7	90	-117.49	-114.67	9.12	**-127.73**	**-124.11**	8.72	-303.72	-286.86	8.64	**-328.12**	**-311.56**	**8.18**	-	-	-
9	3nbm	108	-151.06	-147.89	9.28	**-157.13**	**-156.57**	8.92	-384.07	-370.78	9.433	**-418.60**	**-401.99**	**8.58**	-	-	-
10	3mqo	120	-168.33	-165.13	9.72	**-177.72**	**-174.49**	9.38	-423.92	-396.71	9.44	**-465.74**	**-455.27**	**8.86**	-	-	-
11	3mr0	142	-154.80	-154.80	10.82	**-174.99**	**-170.29**	10.42	-414.49	-396.81	10.32	**-445.33**	**-430.28**	**10.02**	-	-	-
12	3pnx	160	-213.42	-200.33	10.32	**-229.81**	**-221.57**	9.78	-519.38	-474.28	9.765	**-601.23**	**-571.13**	**9.38**	-	-	-

the section, by improvement (degradation) of an approach P over (from) another approach Q, we mean the energy value achieved by Q minus that achieved by P is positive (negative). Note that the energy values themselves are negative.

Improvement Over Other Methods: In Fig. 3(top left), we show the improvement in average energy (ΔE) of the heuristics approach and e_{matrix} over the hybrid approach on 7 proteins for which the hybrid approach is able to produce valid structures. On these proteins, the heuristics approach shows significant improvement in ΔE; the higher, the better. In 2 of the cases (sequence number 2 and 3), the e_{matrix} approach does not show any improvement over the hybrid approach. However, as the length of the protein sequence increases, our algorithm outperforms the hybrid approach.

Effectiveness of Our Approach: In Fig. 3(top right), we show the effectiveness of the heuristics approach over the e_{matrix} approach on cubic and FCC lattices for 12 proteins. In all cases, the heuristics approach shows an improvement in ΔE; the higher, the better. The improvement increases dramatically with the increase of the length of the protein sequence; which is an indication of the scalability. On cubic lattice, the number of neighbor contacts and the degree of freedom is fewer than that of FCC lattice and therefore the energy values are much small in magnitude. However, the improvement level is still significant. In Fig. 3 (bottom left), we show the improvement in average energy (ΔE) of our bounded initialization over the random structured initialization (Dotu et al. 2011) and random valid initialization (Ullah and Steinhöfel 2010) on 7 proteins. On all these proteins, our algorithm initialized by the bounded initialization method produces significantly better structures in comparison to other initialization methods.

Effect of changing move size on stagnation: Jump move size is maintained adaptively with stagnation or change in the global minimum. A plot of change in move size and global minimum against run time for the protein 4rxn is given in Fig. 3 (bottom right). We see that with each new global minimum found, move size is set to the initial value and increased by 1 each time it faces stagnation. The maximum size of the jump move was observed to be 6.

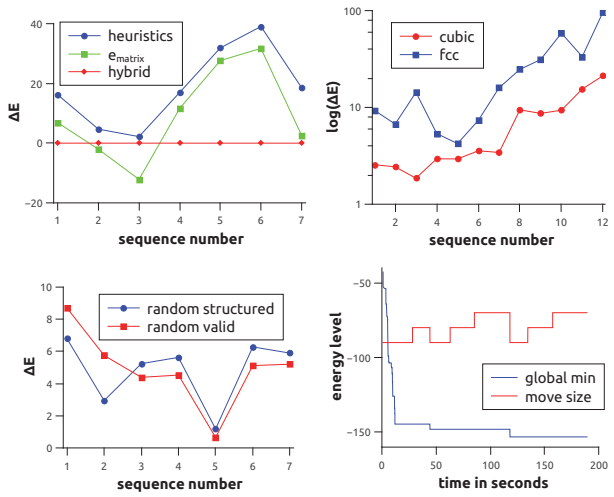

Figure 3: The higher, the better: (top left) improvement in average energy (ΔE) achieved by the e_{matrix} and the heuristics approaches over the hybrid approach on FCC lattice and (top right) log of improvement in average energy achieved ($log\Delta E$) by the heuristics approach over the e_{matrix} approach on cubic and FCC lattices, (bottom left) improvement in average energy by our initialization over the other initialization methods and (bottom right) plot of global minimum energy achieved and move size against time in seconds for the protein 4rxn.

Similarity with Native Structures: Table 1, we also show the Root Mean Square Deviation (RMSD) measures for the conformations produced by the respective algorithms w.r.t. native structures from PDB. For any given structure produced by an algorithm, RMSD $= \sqrt{\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(d_{ij}^{\text{given}}-d_{ij}^{\text{native}})^2}{n*(n-1)/2}}$, where d_{ij}^{given} and d_{ij}^{native} denote the distances between ith and jth amino acids respec-

seq	e_{matrix}	h_1	h_2	h_3	h_4	$h_{1,2}$	$h_{3,4}$	$h_{1,3}$	$h_{1,4}$	$h_{1,3,4}$	$h_{1..4}$	h_b
1	-146.9	-144.2	-73.9	-132.8	-82.7	-107.4	-135.6	-149.6	-116.1	-151.1	-156.3	-64.32
2	-139.9	-135.9	-60.4	-122.1	-73.7	-88.3	-125.5	-143.3	-105.2	-145.2	-146.6	-59.96
3	-184.1	-186.4	-94.1	-166.2	-97.1	-136.6	-169.0	-192.1	-148.4	-193.4	-198.4	-77.45
4	-168.8	-161.1	-83.4	-145.7	-83.2	-107.9	-149.8	-169.8	-126.7	-171.6	-174.1	-75.18
5	-235.7	-226.7	-118.9	-206.6	-122.0	-179.6	-209.1	-235.9	-182.4	-236.7	-239.9	-35.12
6	-196.8	-186.2	-102.1	-168.7	-96.3	-118.3	-171.1	-197.8	-144.3	-198.3	-204.1	-89.66
7	-197.8	-196.8	-83.0	-174.6	-94.6	-126.6	-173.9	-204.4	-148.1	-206.4	-213.8	-74.34

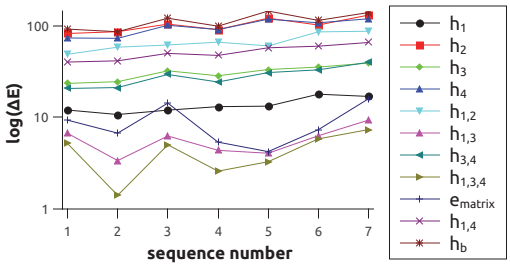

Figure 4: (left) Results for individual and combinations of heuristics and (right) degradation in average energy ($log\Delta E$) for the heuristics over the final approach $h_{1..4}$ with different combinations of heuristics; the higher, the worse.

tively in the given structure and the native structure of the protein. In calculating the RMSD values, the distance between two neighbors in the lattice ($\sqrt{2}$ for FCC and 1 for cubic) is considered to be equal to the average distance (3.8Å) between two α-Carbons on the native structure. The values shown in the table are average RMSD values over the 50 runs for each protein. As we see, our approach also significantly improves over the hybrid approach in terms of the RMSD values; the lower the RMSD score, the better the performance. These values are significant since backbone reconstruction and addition of side-chain atoms can guarantee to produce real protein structures within small (1-2Å) deviation (Rotkiewicz and Skolnick 2008). However, since lattice configurations can only approximate the positions of the amino acids in the real space, lower RMSD values produced by our algorithm are satisfactory.

Performance of the Heuristics Combinations: To see the effect of different heuristics individually and in combinations, we have performed 20 runs of our algorithm for each of the proteins on FCC lattice. Results are shown as a table (left) and a graph (right) in Fig. 4. In the graph, we plot $log\Delta E$ the degradation of average energy (the higher, the worse) of individual and various combinations of the heuristics compared to our final approach $h_{1..4}$ that uses all the heuristics. Thus, the x-axis represents the reference performance of $h_{1..4}$ in the graph.

First, consider the performances when each heuristic is used separately: h_1, h_2, h_3, h_4 are all worse than E, which denotes the performance of the e_{matrix} approach. However, h_1 is significantly better than h_3, and h_3 is significantly better than both h_2 and h_4 while h_4 is slightly better than h_2. These results are explainable because these heuristics are some kind of weak relaxation of the energy function. Heuristics h_1 and h_3 brings respectively the affine or any pairs together while h_2 and h_4 moves respectively repellent or chain neighboring pairs apart from each other. Overall, the performance levels of these individual heuristics appear to be in line with their apparent greediness.

Next, consider the combinations of the heuristics. For space constraints, we only report combinations that are more interesting. When heuristics h_1 and h_3 are used together as they are the best two among the four heuristics under consideration, the combination $h_{1,3}$ produces stronger results than both h_1 and h_3 individually and also stronger than E. We also combine h_4 with h_1; the resulting combination $h_{1,4}$ is better than h_4 but worse than h_1. Similar results are obtained with $h_{1,2}$, the combination of h_1 and h_2. Note that $h_{1,2}$ considers both the attraction-distraction between amino acids. Combination $h_{3,4}$ of h_3 and h_4 performs slightly better than h_3; however, $h_{1,3,4}$ performs significantly better than $h_{1,3}$. Lastly, $h_{1..4}$ (represented by the x-axis) combining all these weak heuristics significantly outperforms all of these combinations along with E. Heuristic $h_{1..4}$, $h_{1,3,4}$ achieve much better energy (improves by 50, 30) values before its first stagnation than e_{matrix} (only 15). In general, $h_{1..4}$ shows more and greater changes in local minima and shorter stagnation periods. We also ran a baseline version (h_b) of our algorithm by randomly selecting the candidates which worked worse than any other combinations.

Conclusion and Future Work

In this paper, we introduced several novel but weak heuristics for low energy protein structure prediction and present a new local search algorithm that strongly exploits these heuristics in a mixed fashion. Our algorithm uses two sets of neighborhood operators such as jump moves and pull moves, and maintains tabu on recent moves. We demonstrated the effectiveness of our approach over a state-of-the-art method and produced results for a 20×20 energy model on both cubic and FCC lattices. Also, our algorithm efficiently produced lower energy structures while other methods failed within a reasonable time limit. In future, we wish to further investigate the suitability of certain heuristics in certain situations. We believe this will lead to a better utilization of the heuristics. Moreover, incorporating the idea of switching the operators depending on the structural properties is another potential research direction. The heuristics that we develop are inspired partly from the domain knowledge and partly from observing the nature of the near optimal structures. This exploration can also be extended to find some other heuristics in protein structure prediction and other domains as well. In future, we also use to explore controlled strategies both in generating neighbors and choosing the best candidate solution at each iteration.

Acknowledgments

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Anfinsen, C. B. 1973. Principles that govern the folding of protein chains. *Science* 181(4096):223–230.

Berrera, M.; Molinari, H.; and Fogolari, F. 2003. Amino acid empirical contact energy definitions for fold recognition in the space of contact maps. *BMC Bioinformatics* 4:8.

Cipra, B. 1998. Packing challenge mastered at last. *Science* 281(5381):1267.

Dotu, I.; Cebrian, M.; Van Hentenryck, P.; and Clote, P. 2011. On lattice protein structure prediction revisited. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* 8(6):1620–1632.

Kapsokalivas, L.; Gan, X.; Albrecht, A. A.; and Steinhöfel, K. 2009. Population-based local search for protein folding simulation in the MJ energy model and cubic lattices. *Computational Biology and Chemistry* 33(4):283–294.

Landau, D., and Binder, K. 2005. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge: Cambridge University Press.

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

Lesh, N.; Mitzenmacher, M.; and Whitesides, S. 2003. 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, 188–195. New York, NY, USA: ACM.

Miyazawa, S., and Jernigan, R. L. 1985. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules* 18(3):534–552.

Newton, M.; Pham, D. N.; Sattar, A.; and Maher, M. J. 2011. Kangaroo: An efficient constraint-based local search system using lazy propagation. In *CP*, 645–659.

Palù, A. D.; Will, S.; Backofen, R.; and Dovier, A. 2004. Constraint based protein structure prediction exploiting secondary structure information. In *Proceedings of Italian Conference on Computational Logic*, CLIC'04.

Palù, A. D.; Dovier, A.; Fogolari, F.; and Pontelli, E. 2011. Exploring protein fragment assembly using CLP. In *IJCAI*, 2590–2595.

Palù, A. D.; Dovier, A.; and Pontelli, E. 2005. Heuristics, optimizations, and parallelism for protein structure prediction in CLP(*FD*). In *PPDP*, 230–241.

Palù, A. D.; Dovier, A.; and Pontelli, E. 2007. A constraint solver for discrete lattices, its parallelization, and application to protein structure prediction. *Softw. Pract. Exper.* 37:1405–1449.

Rashid, M. A.; Hoque, M. T.; Newton, M.; Pham, D. N.; and Sattar, A. 2012. A new genetic algorithm for simplified protein structure prediction. In *Proceedings of the 25th Australian Joint Conference on Artificial Intelligence*, AI'12.

Rotkiewicz, P., and Skolnick, J. 2008. Fast procedure for reconstruction of full-atom protein models from reduced representations. *Journal of Computational Chemistry* 29(9):1460–1465.

Shatabda, S.; Newton, M.; Rashid, M. A.; Pham, D. N.; and Sattar, A. 2013. The road not taken: retreat and diverge in local search for simplified protein structure prediction. *BMC Bioinformatics* 14(2):1–9.

Steinhöfel, K.; Skaliotis, A.; and Albrecht, A. A. 2007. Stochastic protein folding simulation in the d-dimensional HP-model. In *Proceedings of the 1st international conference on Bioinformatics research and development*, BIRD'07, 381–394. Berlin, Heidelberg: Springer-Verlag.

Torres, S. R. D.; Romero, D. C. B.; Vasquez, L. F. N.; and Ardila, Y. J. P. 2007. A novel *ab-initio* genetic-based approach for protein folding prediction. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, 393–400. New York, NY, USA: ACM.

Ullah, A. D., and Steinhöfel, K. 2010. A hybrid approach to protein folding problem integrating constraint programming with local search. *BMC Bioinformatics* 11(S-1):39.