Protein Structure Prediction With Large Neighborhood Constraint Programming Search

Ivan Dotu[1], Manuel Cebrián[1], Pascal Van Hentenryck[1], and Peter Clote[2]

[1] Department of Computer Science, Brown University
Box 1910, Providence, RI 02912
[2] Biology Department, Boston College
Chestnut Hill, MA 02467

Abstract. Protein structure predictions is regarded as a highly challenging problem both for the biology and for the computational communities. Many approaches have been developed in the recent years, moving to increasingly complex lattice models or even off-lattice models. This paper presents a Large Neighborhood Search (LNS) to find the native state for the Hydrophobic-Polar (HP) model on the Face Centered Cubic (FCC) lattice or, in other words, a self-avoiding walk on the FCC lattice having a maximum number of H-H contacts. The algorithm starts with a tabu-search algorithm, whose solution is then improved by a combination of constraint programming and LNS. This hybrid algorithm improves earlier approaches in the literature over several well-known instances and demonstrates the potential of constraint-programming approaches for *ab initio* methods.

1 Introduction

In 1973, Nobel laureat C.B. Anfinsen [2] denatured the 124 residue protein, bovine ribonuclease A, by the addition of urea. Upon removal of the denaturant, the ribonuclease, an enzyme, was determined to be fully functional, thus attesting the successful reformation of functional 3-dimensional structure. Since no chaperone molecules were present, Anfinsen's experiment was interpreted to mean that the native state of a protein is its minimum free energy conformation, and hence that protein structure determination is a computational problem which can in principle be solved by applying a combinatorial search strategy to an appropriate energy model.

Protein structure prediction is historically one of the oldest, most important, yet stubbornly recalcitrant problems of bioinformatics. Solution of this problem would have an enormous impact on medicine and the pharmaceutical industry, since successful tertiary structure prediction, given only the amino acid sequence information, would allow the computational screening of potential drug targets, in that a drug (small chemical ligand) must dock to a complementary portion of the protein surface (such as a G-coupled protein receptor, the most common drug target).[3] of successful drug Indeed, it has been stated that: "Prediction of protein structure *in silico* has thus been the 'holy grail' of computational biologists for many years" [39]. Despite the quantity

[3] The design of HIV protease inhibitors, first described by Lam et al. [29], was based on knowledge of the target structure.

of work on this problem over the past 30 years, and despite the variety of methods developed for structure prediction, no truly accurate *ab initio* methods exist to predict the 3-dimensional structure from amino acid sequence. Indeed, Helles (2008) [24] benchmarked the accuracy of 18 *ab initio* methods, whose average normalized root mean square deviation ranged from 11.17 Å to 3.48 Å, while Dalton and Jackson (2007) [19] similarly benchmarked five well-known homology modeling programs and three common sequence-structure alignment programs. In contrast, computational drug screening requires atomic scale accuracy, since the size of a single water molecule is about 1.4 Å.

In this paper, we describe a combination of constraint programming and Large Neighborhood Search (LNS) to determine close-to-optimal conformations for the Lau-Dill HP-model on the face-centered cubic lattice. Before describing our contribution, we first present an overview of computational methods for protein structure prediction. In general, methods are classified as *homology (comparative) modeling*, *threading*, *lattice model*, and *ab initio*. Protein structure prediction is an immense field that cannot be adequately surveyed in this introduction. Numerous books, such as [50], and excellent reviews, such as [21] are available. Nevertheless, to situate the contribution of our work within the broader scope of protein structure prediction, we briefly describe each of the methods – homology, threading, *ab initio* – in turn, and focus subsequent discussion on lattice models.

In homology (aka comparative) modeling, the amino acid sequence of a novel protein P is aligned against sequences of proteins Q, whose tertiary structure is available in the Protein Data Bank (PDB) [10]. Regions of P aligned to regions of Q are assumed to have the same fold, while non-aligned regions are modeled by interconnecting loops. Examples of comparative modeling software are SWISS-MODEL, developed by M. Peitsch, T. Schwede et al., and recently described in [3], as well as MODELER developed by the Šali Lab [26]. Comparative modeling relies on the assumption that evolutionarily related (homologous) proteins retain high sequence identity and adopt the same fold.

Threading [40, 31], though known to be NP-complete [30], is a promising *de novo* protein structure approach, which relies on *threading* portions $a_i, \ldots, a_j$ of the amino acid sequence $a_1, \ldots, a_n$ onto a *fragment library*, which latter consists of frequently adopted partial folds. Pseudo-energy (aka knowledge-based potential) is computed from the frequency of occurrence of certain folds with certain types of amino acid sequence. Impressive results have been obtained with the Skolnick Lab program I-TASSER [47] with web server [51], which yielded the best-ranked structure predictions in the blind test CASP-7 (Critical Assessment of Techniques for Protein Structure Prediction) in 2006. Success of threading hinges on two things: *energetics*, i.e., that the PDB is relatively saturated and contains occurrences of almost all protein folds, and *search strategy*, i.e., usually Monte-Carlo or some type of branch-and-bound algorithm. According to a study of Zhang and Skolnick [52], the PDB is currently sufficiently saturated to permit adequate threading approaches, albeit with insufficient accuracy for the requirements of computational drug design.[4]

[4] According to [52], using the TASSER algorithm, "in 408 cases the best of the top five full-length models has a RMSD < 6.5 Ångstroms."

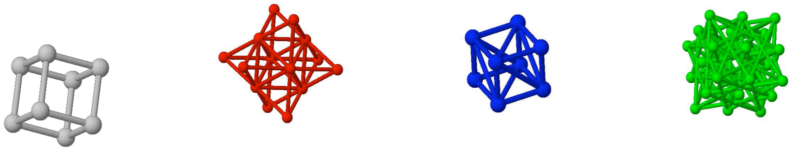


Fig. 1. Lattices used in protein structure modeling. *(a)* Points (x, y, z) in cubic lattice, satisfying $0 \leq x, y, z \leq 1$. *(b)* Points (x, y, z) in FCC lattice, satisfying $0 \leq x, y, z \leq 2$. *(c)* Points (x, y, z) in tetrahedral lattice, satisfying $0 \leq x, y, z \leq 1$. *(d)* Points (x, y, z) in 210 (knight's move) lattice, satisfying $0 \leq x, y, z \leq 2$.

Despite advances in comparative modeling and threading, there is an interest in *ab initio* protein structure prediction, since this is the only method that attempts to understand protein folding from basic principles, i.e., by applying a search strategy with (generally) a physics-based energy function. Moreover, only *ab initio* methods can be applied for proteins having no homology with proteins of known structure. In molecular dynamics (MD), protein structure is predicted by iteratively solving Newton's equations for all pairs of atoms (possibly including solvent) using mean force potentials, that generally include pairwise (non-contact) terms for Lennard-Jones, electrostatic, hydrogen bonding, etc. Well-known MD software CHARMM [14] and Amber [20], as well as variant Molsoft ICM [1], the latter employing internal coordinates (dihedral angle space) and local optimization, are used to simulate protein docking, protein-ligand interactions, etc. since molecular dynamics generally is too slow to allow *ab initio* folding of any but the smallest proteins. Other *ab initio* methods include the Baker Lab program Rosetta [12], benchmarked in [24] with comparable accuracy as the Skolnick Lab program I-TASSER [47]. Search strategies of *ab initio* methods include molecular dynamics simulation, Metropolis Monte-Carlo (Rosetta [12]), Monte-Carlo with replica exchange (I-TASSER [47]), branch-and-bound (ASTROFOLD [27]), integer linear programming (ASTROFOLD [27]), Monte-Carlo with simulated annealing, evolutionary algorithms, and genetic algorithms.

2 Problem Formalization

A *lattice* is a discrete integer approximation to a vector space, formally defined to be the set of *integral* linear combinations of a finite set of vectors in $\mathbb{Z}^n$; i.e.,

$$L = \left\{ \sum_{i=1}^{k} a_i \boldsymbol{v}_i : a_i \in \mathbb{Z} \right\} \tag{1}$$

where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{Z}^n$. If k is the minimum value for which (1) holds, then $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ form a *basis*, and k is said to be the *dimension* (also called *coordination* or *contact* number) of L. Two lattice points $p, q \in L$ are said to be *in contact* if $q = p + \boldsymbol{v}_i$ for some vector $\boldsymbol{v}_i$ in the basis of L. Historically, many different lattices have been considered, some of which are depicted in Figure 1. For more details on properties of these and other lattices, see the book by Conway and Sloane [16]. In this paper, we

	H	P
H	-1	0
P	0	0

	H	P	N	X
H	-4	0	0	0
P	0	+1	-1	0
N	0	-1	+1	0
X	0	0	0	0

Fig. 2. Energy for HP- and HPNX-model.

consider the face-centered cubic (FCC) lattice which is generated by the following 12 basis vectors (identified with compass directions [46]):

$$\begin{array}{lll}
N:(1,1,0) & S:(-1,-1,0) & W:(-1,1,0) \\
E:(1,-1,0) & NW_{+}:(0,1,1) & NW_{-}:(0,1,-1) \\
NE_{+}:(1,0,1) & NE_{-}:(1,0,-1) & SE_{+}:(0,-1,1) \\
SW_{+}:(-1,0,1) & SE_{-}:(0,-1,-1) & SW_{-}:(-1,0,-1).
\end{array}$$

It follows that the FCC lattice consists of all integer points (x, y, z), such that $(x+y+z) \bmod 2 = 0$, and that lattice points $p = (x, y, z)$ and $q = (x', y', z')$ are in *contact*, denoted by $co(p, q)$, if $(x - x') + (y - y') + (z - z') \bmod 2 \equiv 0$, $|x - x'| \leq 1$, $|y - y'| \leq 1$, and $|z - z'| \leq 1$. We will sometimes state that lattice points p, q are at *unit distance*, when we formally mean that they are in contact. Since the distance between two successive alpha-carbon atoms is on average 3.8Å with a standard deviation of 0.04Å, a reasonable coarse-grain approach is to model an n-residue protein by a self-avoiding walk $p_1, \ldots, p_n$ on a lattice.

In 1972, Lau and Dill [32] proposed the *hydrophobic-hydrophilic* (HP) model, which provides a coarse approximation to the most important force responsible for the hydrophobic collapse which has been experimentally seen in protein folding. Amino acids are classified into either hydrophobic (e.g. Ala, Gly, Ile, Leu, Met, Phe, Pro, Trp, Val) or hydrophilic (e.g. Arg, Asn, Asp, Cys, Glu, Gln, His, Lys, Ser, Thr, Tyr) residues. In the HP-model, there is an energy of -1 contributed by any two non-consecutive hydrophobic residues that are *in contact* on the lattice. For this reason, the HP-model is said to have a contact potential, depicted in the left panel of Figure 2, where 'H' designates hydrophobic, while 'P' designates polar (i.e., hydrophilic). To account for electrostatic forces involving negatively charged residues (Asp, Glu) and positively charged residues (Arg, His, Lys), the HP-model has been extended to the HPNX-model, with hydrophobic (H), positively charged (P), negatively charged (N) and neutral hydrophilic (X) terms. The right panel of Figure 2 depicts the HPNX-potential used in [11].

Though Lau and Dill [32] originally considered only the 2-dimensional square lattice, their model allowed the formulation of the following simply stated combinatorial problem. For a given lattice and an arbitary HP-sequence, determine a self-avoiding walk on the lattice having minimum energy, i.e., a minimum energy lattice conformation. This problem was shown to be NP-complete for the 2-dimensional square lattice by [17] and for the 3-dimensional cubic lattice by Berger and Leighton [9].

3 Related Work

Approaches to the HP Model We first survey some search strategies for the HP-model. In [48], Yue and Dill applied "constraint-based exhaustive search"[5] to determine the minimum energy conformation(s) of several small proteins including crambin, when represented as HP-sequences on the cubic lattice. Necessarily, any exhaustive search is limited to very small proteins, since the number of conformations for an n-mer on the 3-dimensional cubic lattice is estimated to be approximately 4.5^n [33]. In [43], Unger and Moult described a genetic algorithm for the HP-model on the 2-dimensional square lattice, where pointwise mutation corresponds to a conformation pivot move. This approach was extended in Backofen, Will, and Clote [6] to a genetic algorithm on the FCC lattice, in order to quantify hydrophobicity in protein folding.

In [7, 4], Backofen applied constraint programming to the HP-model and Backofen and Will applied constraint programming to the HPNX-model [5], thus providing an exact solution for small HP- and HPNX-sequences beyond the reach of exhaustive methods. In [45, 8], Will and Backofen precomputed *hydrophobic cores*, maximally compact face-centered cubic self-avoiding walks of (only) hydrophobic residues. By threading an HP-sequence onto hydrophobic cores, the optimum conformation could be found for certain examples; however, if threading is not possible (which is often the case), no solution is returned.

Dal Palu et al. [18] use secondary structure and disulfide bonds used as constraints using constraint logic programming over finite domains to compute a predicted structure on the face-centered cubic lattice. They describe tests ranging from the 12 residue fragment (PDB code 1LE0) with RMSD of 2.8 Å achieved in 1.3 seconds, to the 63 residue protein (PDB code 1YPA) with RMSD of 17.1 Å in 10 hours. Further optimization was performed after the alpha-carbon trace was replaced by an all-atom model (presumably using well-known Holm-Sander method [25]), thus achieving an all-atom prediction of the 63 residue protein (PDB code 1YPA) with RMSD of 9.2 Å within 116.9 hours computation time. This study suggests that protein structure prediction might best proceed in a hierarchical fashion, first taking into account secondary structure on a coarse-grain lattice model and subsequently performing all-atom refinement.

Beyond the HP Model The HP-model can be viewed as a coarse approximation of more complex *contact potentials*. In [35], Miyazawa and Jernigan introduced two kinds of contact potential matrices, i.e., 20×20 matrices that determine a residue-dependent energy potential to be applied in the case that two residues are in contact (either on the lattice, or within a fixed threshold such as 7 Åfrom each other). Recently, Pokarowski et al. [37] analyzed 29 contact matrices and showed that in essence all known contact potentials are one of the two types they introduced in [35]. Their first contact potential is given by the formula $e(i, j) = h(i) + h(j)$, where $1 \leq i \leq 20$ ranges over the 20 amino acids and h is a residue-type dependent factor that is highly correlated with frequency of occurrence of a given amino acid type in a non-redundant collection of proteins. Their second contact potential is given by the formula $e(i, j) = c_0 - h(i)h(j) + q(i)q(j)$, where c_0 is a constant, h is highly correlated with the Kyte-Doolittle hydrophobicity

[5] Despite the name, the method of Yue and Dill did not did not involve constraint programming.

scale [28], and a residue-type dependent factor q is highly correlated isoelectric points pI. The “knight’s move” 210 lattice was used by Skolnick and Kolinski [41] to fold the 99-residue beta protein, apoplastocyanin, to within 2 Åof its crystal structure with PDB accession code 2PCY.

4 Why Constraint Programming?

Our earlier work [13] applied a tabu-search algorithm to obtain approximate solutions for protein folding for the HP-model on FCC lattice. The goal of this paper is to evaluate a similar model using a large neighborhood search based on constraint programming, both to improve earlier results and to assess their quality. The improvements obtained by the CP-based LNS indicate that this approach provides significant benefits over a pure local search algorithm. More generally, as explained in the introduction, protein structure prediction can be viewed as the application of a search engine (Monte-Carlo, Monte-Carlo with replica exchange, genetic algorithm, integer programming, ...) to a physics-based or knowledge-based energy function. This paper evaluates CP-Based large neighborhood search on the Harvard instances, a standard benchmark for assessing accuracy of structure prediction for the HP-model. Our successful application of LNS to the face-centered cubic lattice suggests the potential of using this constraint-programming strategy in a hierarchical manner with successive refinements to perform all-atom structure prediction – a task for future research.

5 The Implementation

5.1 The CP Model

The CP model receives as input a sequence of binary values H_i $(0 \leq i < n)$ denoting whether aminoacid i is hydrophobic ($H_i = 1$). Its output associates each aminoacid i with a point (x_i, y_i, z_i) in the FCC lattice. Recall that the FCC lattice is the closure of 12 vectors $V = \{v_0, \ldots, v_{11}\}$ defined as follows:

$$\begin{array}{llll} v_0 = \{1,1,0\} & v_1 = \{-1,-1,0\} & v_2 = \{-1,1,0\} & v_3 = \{1,-1,0\} \\ v_4 = \{1,0,1\} & v_5 = \{-1,0,-1\} & v_6 = \{-1,0,1\} & v_7 = \{1,0,-1\} \\ v_8 = \{0,1,1\} & v_9 = \{0,-1,-1\} & v_{10} = \{0,-1,1\} & v_{11} = \{0,1,-1\}. \end{array}$$

Decision Variables Although the output of the model maps each aminoacid i into a FCC lattice point, the model uses move vectors as decision variables. These vectors (m_i^x, m_i^y, m_i^z) specify how to move from point $i - 1$ to point i in the self-avoiding walk. The use of *move variables* greatly simplifies the problem statement.

The Domain Constraints Each move variable (m_i^x, m_i^y, m_i^z) has a finite domain consisting of the FCC lattice vectors $\{v_0, \ldots, v_{11}\}$, that is

$$(m_i^x, m_i^y, m_i^z) \in \{v_0, \ldots, v_{11}\}.$$

Each coordinate x_i, y_i, and z_i in the 3D point (x_i, y_i, z_i) associated with aminoacid i has a finite domain $0..2n$.

The Lattice Constraints The lattice constraints link the *move variables* and the points in the FCC lattice. They are specified as follows:

$$\forall\, 0 < i < n : \; x_i = x_{i-1} + m_i^x \;\&\; y_i = y_{i-1} + m_i^y \;\&\; z_i = z_{i-1} + m_i^z .$$

The model also uses the redundant constraints

$$(x_i + y_i + z_i) \text{ mod } 2 = 0$$

which are implied by the FCC lattice. In addition, the initial point is fixed.

The Self-Avoiding Walk Constraints To express that all aminoacids are assigned different points in the FCC lattice, the model uses a constraint

$$abs(\sum_{k \in i..j} m_k^x) + abs(\sum_{k \in i..j} m_k^y) + abs(\sum_{k \in i..j} m_k^z) \neq 0$$

for each pair (i, j) of aminoacids, ensuring the moves from the position of aminoacid i do not place j at the same position as i. Indeed, the two points (x_i, y_i, z_i) and (x_j, y_j, z_j) are at the same position if each of the sums in the above expression is zero.

The Objective Function The objective function maximizes the number of contacts between hydrophobic aminoacids

$$\sum_{i,j \mid i+1<j} (d_{ij} = 2) \times H_i \times H_j$$

where d_{ij} denotes the square of Euclidean distance between aminoacids i and j, i.e.,

$$d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 .$$

Since the minimal distance in the FCC lattice is $\sqrt{2}$, the condition $d_{ij} = 2$ holds when there exists a contact between aminoacids i and j.

5.2 The Search Procedure

The search procedure assigns positions to the aminoacids in sequence by selecting moves in their domains. The only heuristic choice thus concerns which moves to select, which uniquely determines the position of the next aminoacid. In the course of this research, a number of move selection heuristics were evaluated. Besides the traditional lexicographic and random value selections, the heuristics included

1. **Minimizing the distance to the origin:** Choosing the move minimizing the distance of the corresponding aminoacid to the origin.
2. **Minimizing the distance to the centroid:** Choosing the move minimizing the distance of the corresponding aminoacid to the centroid.
3. **Maximizing density:** Choosing the move maximizing the density of the structure.

4. **Maximizing hydrophobic density:** Choosing the move that maximizing the density of the structure consisting only of the hydrophobic aminoacids.

Note that the centroid of the conformation is defined as

$$(\frac{1}{n}\sum_{i=0}^{n-1} x_i, \frac{1}{n}\sum_{i=0}^{n-1} y_i, \frac{1}{n}\sum_{i=0}^{n-1} z_i).$$

Most of the dedicated heuristics bring significant improvements in performance, although those minimizing the distance to the origin and the centroid seem to be most effective. Our implementation randomly selects one of the two heuristics.

5.3 Strengthening the Model During Search

We now describe a number of tightenings of the model which are applied during search. Their main benefit is to strengthen the bound on the objective function.

Linking FCC Moves and Distance Constraints In the model described so far, the distance between two aminoacids ignores the fact that the points are placed on the FCC lattice. The model may be improved by deriving the fact that two aminoacids are necessarily placed at a distance greater than $\sqrt{2}$ and thus cannot be in contact. Such derived information directly improves the bound on the objective function.

However computing the possible distances between two aminoacids is quite complex in general. As a result, our constraint-programming algorithm only generates relevant distances each time a new aminoacid is positioned. More precisely, assuming that aminoacid i has just been positioned on the FCC lattice, the algorithm determines which unassigned aminoacids cannot be in contact with already placed aminoacids (only for H-type aminoacids). The key idea is to compute the shortest path sp_{ij} in the FCC lattice between aminoacid i and an already placed aminoacid j: It then follows that unassigned aminoacids $i+1, \ldots, i+sp_{ij}-2$ cannot be in contact with j. Formally, after placing aminoacid i, the model is augmented with the constraints

$$\forall 0 \leq j \leq i-2, i+1 \leq l \leq i+sp_{ij}-2 : d_{jl} > 2$$

which ensures that aminoacids j and l cannot be in contact.

Bounding the Number of Contacts The expression of the objective function also does not take into account how the aminoacids are placed in the FCC lattice. As a result, it typically gives weak bounds on the objective value. This section shows how to bound the objective value at a search node more effectively.

The key idea to bound the objective value is to compute the maximum number of contacts for each unassigned aminoacid independently, thus ignoring their interactions through the self-avoiding walk. Consider a node of search tree where the sequence can be partitioned into the concatenation $A :: U$, where A is the subsequence of already positioned aminoacids in which i is the last assigned one (also, we only consider $a \in A \| H_a == 1$ and $k \in U \| H_k == 1$). The objective function can then be bounded by

$$obj \leq contact(A) + \sum_{kinU} min(maxContact(k), bcontact(k, A) + fcontact(k, U))$$

where $contact(A)$ denotes the number of contacts in subsequence A, $bcontact(k, A)$ bounds the number of contacts of an aminoacid $k \in U$ with those aminoacids in A, and $fcontact(k, U)$ bounds the number of contacts of k with those aminoacids in U occurring later in the sequence. The contacts of each aminoacid $k \in U$, $maxContact(k)$, are bounded by 10, since a point in the FCC lattice has 12 neighbors and there cannot be any contact between two successive aminoacid in the sequence. However, if $k == n - 1$, i.e., if k is the last aminoacid of the sequence then $maxContact(k) == 11$, since that k has no successor aminoacid.

To bound the contact of aminoacid k with A, the idea is to consider the neighbors of each aminoacid $a \in A$ and to find the one maximizing the contacts with k, i.e.,

$$\begin{array}{ll} bcontact(k, A) & = \max_{a \in A} bcontact(k, a, A) \\ bcontact(k, a, A) & = \#\{j \in A \mid j \in N(a) \wedge j \in R(k, a)\}. \end{array}$$

where $N(a)$ denotes the neighbors of aminoacid a and $R(k, a)$ denotes the aminoacid in A reachable from k, i.e.,

$$R(k, A) = \{a \in A \mid sp_{ai} \leq (k - i) + 1\}.$$

Recall that i is the last aminoacid assigned. Finally, to bound the number of contacts of k with those aminoacids occurring later in the sequence, we use

$$fcontact(k, U) = \sum_{l \in U : l \geq k+2} H_l$$

to count the number of hydrophobic aminoacids occurring later in U that can be in contact with k.

This bound can be computed in time $O(n^2)$ and is quite tight when the number of aminoacids in U is reasonably small.

5.4 Large Neighborhood Search

Structure prediction is a highly complex combinatorial optimization problem. As a result, constraint programming search may spend considerable time in suboptimal regions of the search space. To remedy this limitation, our algorithm uses the idea of large neighborhood search (LNS) [38] which focuses on reoptimizing subparts of a solution. Given a feasible walk σ, the idea is to solve the structure prediction problem for a subsequence of the original sequence, assuming that the remaining aminoacids are positioned like in σ. More precisely, given an interval $i..j$, an LNS optimization step consists of solving the original model with the additional constraints

$$\forall\, k : 0 \leq k < i : x_i = \sigma(x_i) \ \wedge \ y_i = \sigma(y_i) \ \wedge \ z_i = \sigma(z_i)$$

and

$$\forall\, k : j < k < n : x_i = \sigma(x_i) \ \wedge \ y_i = \sigma(y_i) \ \wedge \ z_i = \sigma(z_i)$$

where $\sigma(x)$ denotes the value of variable x in solution σ.

```
1.  **LNS_PSP**$(\sigma)$
2.      $limit \leftarrow limit_0$
3.      $fraction \leftarrow fraction_0$
4.      **for** $m$ iterations **do**
5.          **uniform select** $i \in 1..n-1$
6.          $size \leftarrow n \cdot fraction$
7.          $j \leftarrow i + size$
8.          $\langle \sigma^*, explored \rangle$ = **CPSolve**$(\sigma, i..j, limit)$
9.          **if** $\sigma^* \neq \perp$ **then**
10.             $\sigma \leftarrow \sigma^*$
11.             $limit \leftarrow limit_0$
12.             $fraction \leftarrow fraction_0$
13.         **else if** *explored* **then**
14.             $fraction \leftarrow fraction + \Delta fraction$
15.         **else**
16.             $limit \leftarrow limit + \Delta limit$
17.     **return** $\sigma$
```

Fig. 3. LNS for Protein Structure Prediction ($limit_0$=500 failures, $fraction_0 = \frac{3}{100}$, $\Delta fraction = \frac{1}{1000}$ and $\Delta limit$=100 failures).

The complete LNS algorithm is depicted in Figure 3. It receives as input a high-quality solution produced by the tabu-search algorithm described in [13] and uses a subroutine *CPSolve*$(\sigma, i..j, l)$ which solves augmented models using constraint programming and terminates after at most $limit$ failures had occurred or when the entire search space has been explored. It returns a pair $\langle \sigma^*, explored \rangle$, where σ^* is either a new best solution or $\perp$ if no such solution was found, and *explored* is a boolean which is true when the entire search space has been explored for the augmented model. Lines 2–3 initialize two parameters: the limit on the number of failures and the fraction of the subsequence to (re)-position on the FCC lattice. Line 8 is the call to the constraint-programming solver. After this call there are three possibilities. First, that the search is successful: then the best solution is updated and the parameters are re-initialized (lines 9–12). Second, that the search space has been explored entirely with no improvement; the fraction of the sequence to re-position is increased at a certain rate $\Delta fraction$ (lines 13–14). Finally, that *CPSolve* reached $limit$ without an improvement: the number of failures is increased in $\Delta limit$ to give it more time for to succeed in the next trial (lines 15–16).

6 Experimental Results

All the results presented in this section have been produced by a COMET [44, 34] implementation of the LNS algorithm, run on a single core of a 60 Intel based, dual-core, dual processor, Dell Poweredge 1855 blade server. Each blade has 8G of memory and a 300G local disk, and each execution was carried out on a single core. Each of the considered benchmarks was run for about 48 hours.

Seq.	Lowest LS E	median time	Lowest LNS E	time	Improvement %
H1	-68	114 sec.	-69	5.32 sec.	1.47
H2	-69	265 sec.	Not improv.		0
H3	-68	72 sec.	-71	28.64 sec.	4.41
H4	-66	44 sec.	-69	26.55 sec.	4.55
H5	-66	53 sec.	-67	4.18 sec.	1.52
H6	-70	149 sec.	Not improv.		0
H7	-68	8 sec.	-69	9.86 sec.	1.47
H8	-64	10 milisec.	-65	18.3 sec.	1.56
H9	-69	89 sec.	Not improv.		0
H10	-66	30 sec.	-67	9.74 mins.	1.52

Table 1. Results for the Harvard instances.

Seq.	Native E	Lowest LS E	median time	Lowest LNS E	time	Improvement %
S1	-357	-325	15.98 min.	-346	1.61 hour	6.46
S2	-360	-315	19.18 min.	-343	4.48 hours	8.89
S3	-367	-307	1.14 min.	-341	54.18 mins.	11.07
S4	-370	-318	13.14 min.	-340	7.4 hours	6.92
R1	-384	-284	2.09 min.	-337	1.3 hours	18.66
R2	-383	-290	18.8 min.	-325	7.67 hours	12.07
R3	-385	-282	6.45 min.	-317	2.08 hours	12.41

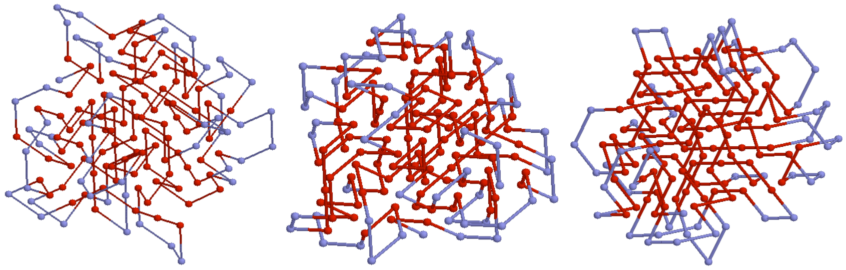
Table 2. Results for the Will's instances.

6.1 The Harvard Instances

Reference [49] contains a comparison of several methods to fold 10 different proteins, called the ”Harvard instances”, on the cubic lattice. The cubic lattice has been heavily studied as pointed out in the introduction, but the FCC lattice has been shown to admit the tightest packing of spheres [15], indicating that it allows for more complex 3D structures. The first results for these instances on the FCC lattice were presented in [13] and confirmed that the FCC lattice allows for structures with much lower energy than the cubic lattice. Table 1 depicts the results of our hybrid algorithm, starting with a local-search algorithm and improving the result with LNS. Note that the energy shown in the table corresponds to minus the number of HH contacts. The LNS step improves 7 out of 10 solutions quickly. Since no complete search algorithms have been applied to these instances on the FCC lattice, the energy of the optimal structure is not known. However, given the consistency in the energies of all the sequences (which all have 48 aminoacids and 24 hydrophobic aminoacids), it is probably the case that these results are near-optimal.

6.2 Other Instances

We also evaluated our algorithm with the only FCC foldings available in the literature. Table 2 depicts a comparison for 7 instances found in [46]. All instances contain 100 H aminoacids, and the R instances have a total of 200 aminoacids, while the S instances



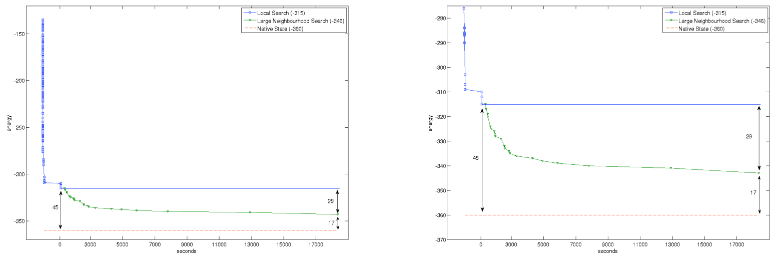
(a) Lowest E. Config. (LS) (b) Low. E. Con. (LS+LNS) (c) Native Configuration

Fig. 4. Lowest energy configurations achieved for Will's instance S2.

range between 130 and 180 aminoacids. Table 2 also shows optimal energies for these instances.[6] The results demonstrate that LNS significantly improves the local search algorithm, with improvements ranging from 6% to 18%. The largest improvements occur on the R instances, which is explained by the lower quality of local search for these instances. The results on the S instances are within 8% of the optimal solution, while the algorithm is within 18% of the optimal solutions on the R instances. Figure 4 depicts a 3D view of the best configuration found for S2 for the local search in [13], the LNS algorithm, and the native state.

It is also important to stress how the optimal solutions were obtained in [46]. Will's algorithm solves a substantially different problem which consists of threading a sequence into a pre-calculated H core. The algorithm relies on a set of precomputed (optimal and suboptimal) cores and tries to map the protein on these cores. Such threading for the protein may not exist for any of these cores or may not be found within the given time limit, in which case the threading algorithm may not provide any solution. There is thus a fundamental conceptual difference between the algorithm presented in this paper and the hydrophobic-core constraint-programming method of Will and Backofen [45, 8], which can be captured using the concepts of Monte-Carlo and Las Vegas algorithms from theoretical computer science [36]. Monte-Carlo algorithms always converge, but have a (small) probability of error in the solution proposed; in contrast, Las Vegas algorithms always return the correct solution, but have a (small) probability of not converging. By analogy, our approach (LNS with constraint programming) is akin to a Monte-Carlo method, in that an approximate solution is always returned. In contrast, hydrophobic-core constraint programming is akin to a Las Vegas method, in that any solution returned is an exact (optimal) solution; however, in many cases, the hydrophobic-core method fails to return any answer. Reference [46, p. 129] includes a table indicating that the threading algorithm only solves 50% of the instances with an H core of size 100 within the given time limit. The instances for which they report a solution are those which can be threaded in an optimal H core. These instances are heavily biased against our algorithm and none of the other sequences are available. Thus, a fair comparison of the algorithms is not possible at this stage, since only the above 7 sequences are available and they belong to the 50% the threading algorithm can solve.

[6] Personal Communication with Sebastian Will.



(a) Behaviour over 5 hours of LS + LNS

(b) Zoom on LNS behaviour

Fig. 5. Algorithm Behavior over Time for Will's instance S2.

It is also important to mention that Will's algorithm relies heavily on the definition of energy and it is hard to generalize to other energy models. Our algorithm solves the problem *ab-initio* and has the potential of obtaining near-optimal solution for general proteins. In addition, our approach is completely general and may encompass different notions of energies at very small cost of implementation. Moreover, some preliminary results indicate that it can be applied to problems such as RNA structure prediction with minimal modifications.

Finally, figure 5 depicts the improvement of the solutions of our algorithm over time. The algorithm exhibits a steep descent, followed by a long plateau, and then another steep descent. It is interesting to see how the local search, the LNS (on their own) and the complete process (local search + LNS), they all present the same behavior.

7 Conclusions and Future Work

This paper presented an LNS algorithm for finding high-quality self avoiding walk for the Hydrophobic-Polar (HP) energy model on the Face Centred Cubic (FCC) lattice. The algorithm relies on a local search initial solution which is then improved by a constraint-programming LNS strategy. Experimental results on the standard Harvard instances show improvements over previously presented results, while significant improvements are achieved in other larger instances. The result shows work shows that the hybridization of local search and constraint programming has great potential to approach the highly combinatorial problem of structure prediction.

References

1. R.A. Abagyan, M.M. Totrov, and D.A. Kuznetsov. ICM: a new method for structure modeling and design: Applications to docking and structure prediction from the distorted native conformation. *J. Comp. Chem.*, 15:488–506, 1994.
2. C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.
3. K. Arnold, L. Bordoli, J. Kopp, and T. Schwede. The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, January 2006.
4. R. Backofen. The protein structure prediction problem: A constraint optimization approach using a new lower bound. *Constraints*, 6(2-3):223–255, 2001.
5. R. Backofen, S. Will, and E. Bornberg-Bauer. Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. *Bioinformatics*, 15(3):234–242, March 1999.
6. R. Backofen, S. Will, and P. Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. *Pacific Symposium on Biocomputing*, 5:92–103, 2000.
7. Rolf Backofen. Using constraint programming for lattice protein folding. In *Workshop on Constraints and Bioinformatics/Biocomputing*, 1997. Held in conjunction with *Third International Conference on Principles and Practice of Constraint Programming (CP97)*.
8. Rolf Backofen and Sebastian Will. A constraint-based approach to structure prediction for simplified protein models that outperforms other existing methods. In *Proceedings of the 19th International Conference on Logic Programming (ICLP 2003)*, pages 49–71, 2003.
9. B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (hp) model is NP-complete. *Journal of Computational Biology*, 5:27–40, 1998.
10. H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J. D. Westbrook, and C. Zardecki. The Protein Data Bank. *Acta Crystallogr. D. Biol. Crystallogr.*, 58(Pt):899–907, June 2002.
11. E. Bornberg-Bauer. Chain growth algorithms for HP-type lattice proteins. In *RECOMB*, pages 47–55. ACM Press, 1997.
12. P. Bradley, K. M. Misura, and D. Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, September 2005.
13. M. Cebrian, I. Dotu, P. Van Hentenryck and Peter Clote. Protein Structure Prediction on the Face Centered Cubic Lattice by Local Search. *To appear in AAAI'08*.
14. B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4:187–217, 1983.
15. B. Cipra. Packing challenge mastered at last. *Science*, 281:1267, 1998.
16. J.H. Conway and N.J.A. Sloane. *Sphere Packing, Lattices and Groups*. Springer-Verlag, 1998. Third edition.
17. P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *J. Comp. Biol.*, 5(3):523–466, 1998.
18. A. Dal Palu, A. Dovier, and F. Fogolari. Constraint Logic Programming approach to protein structure prediction. *BMC. Bioinformatics*, 5:186, November 2004.
19. J. A. Dalton and R. M. Jackson. An evaluation of automated homology modelling methods at low target template sequence similarity. *Bioinformatics*, 23(15):1901–1908, August 2007.
20. Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. Wang, and P. Kollman. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *J. Comput. Chem.*, 24(16):1999–2012, December 2003.
21. C. A. Floudas. Computational methods in protein structure prediction. *Biotechnol. Bioeng.*, 97(2):207–213, June 2007.
22. N. Go and H. Taketomi. Respective roles of short- and long-range interactions in protein folding. *Proc. Natl. Acad. Sci. U.S.A.*, 75(2):559–563, February 1978.
23. N. Go and H. Taketomi. Studies on protein folding, unfolding and fluctuations by computer simulation. III. Effect of short-range interactions. *Int. J. Pept. Protein. Res.*, 13(3):235–252, March 1979.
24. G. Helles. A comparative study of the reported performance of ab initio protein structure prediction algorithms. *J. R. Soc. Interface*, 5(21):387–396, April 2008.
25. L. Holm and C. Sander. Database algorithm for generating protein backbone and side-chain co-ordinates from a C $alpha$ trace application to model building and detection of co-ordinate errors. *J. Mol. Biol.*, 218(1):183–194, March 1991.
26. B. John and A. Sali. Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic. Acids. Res.*, 31(14):3982–3992, July 2003.
27. J.L. Klepeis and C.A. Floudas. Prediction of β-sheet topology and disulfide bridges in polypeptides. *Journal of Computational Chemistry*, 24(2):191–208, 2002.
28. J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, 157(1):105–132, May 1982.
29. P. Y. Lam, P. K. Jadhav, C. J. Eyermann, C. N. Hodge, Y. Ru, L. T. Bacheler, J. L. Meek, M. J. Otto, M. M. Rayner, Y. N. Wong, et al. Rational design of potent, bioavailable, nonpeptide cyclic ureas as HIV protease inhibitors. *Science*, 263(5145):380–384, January 1994.
30. R. H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein. Eng.*, 7(9):1059–1068, September 1994.
31. R.H. Lathrop and T.F. Smith. Global optimum protein threading with gapped alignment and empirical pair score functions. *J. Mol. Biol.*, 255(4):641–665, 1996.
32. K.F. Lau and K. A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Journal of the American Chemical Society*, 22:3986–3997, 1989.

33. N. Madras and G. Slade. *The Self-Avoiding Walk*. Birkh?user, Boston, 1996. Series: Probability and its Applications, 448 p., ISBN: 978-0-8176-3891-7.
34. L. Michel, A. See and P. Van Hentenryck Parallelizing Constraint Programs Transparently. *CP'2007*, Providence, RI, 2007.
35. S. Miyazawa and R. L. Jernigan. Self-consistent estimation of inter-residue protein contact energies based on an equilibrium mixture approximation of residues. *Proteins.*, 34(1):49–68, January 1999.
36. C. Papadimitriou. Computational Complexity, Addison Wesley, 1994.
37. P. Pokarowski, A. Kloczkowski, R. L. Jernigan, N. S. Kothari, M. Pokarowska, and A. Kolinski. Inferring ideal amino acid interaction forms from statistical protein contact potentials. *Proteins.*, 59(1):49–57, April 2005.
38. P. Shaw Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. *CP'98*, 1998.
39. N. Siew and D. Fischer. Convergent evolution of protein structure prediction and computer chess tournaments: CASP, Kasparov, and CAFASP. *IBM Systems Journal*, 40(2):410–425, 2001.
40. M. Sippl. Calculation of conformation ensembles from potentials of mean force. *J. Mol. Biol.*, 213:859–883, 1990.
41. J. Skolnick and A. Kolinski. Simulations of the Folding of a Globular Protein. *Science*, 250(4984):1121–1125, November 1990.
42. H. Taketomi, F. Kano, and N. Go. The effect of amino acid substitution on protein-folding and -unfolding transition studied by computer simulation. *Biopolymers.*, 27(4):527–559, April 1988.
43. R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231:75–81, 1993.
44. P. Van Hentenryck and L. Michel Constraint-Based Local Search. *The MIT Press*, 2005.
45. S. Will. Constraint-based hydrophobic core construction for protein structure prediction in the face-centered-cubic lattice. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 7, pages 661–672, 2002. World Scientific Publishing Co., Singapore.
46. Sebastian Will. *Exact, Constraint-Based Structure Prediction in Simple Protein Models*. PhD thesis, Friedrich-Schiller-Universität Jena, April 2005.
47. S. Wu, J. Skolnick, and Y. Zhang. Ab initio modeling of small proteins by iterative TASSER simulations. *BMC. Biol.*, 5:17, 2007.
48. K. Yue and K. A. Dill. Folding proteins with a simple energy function and extensive conformational searching. *Protein. Sci.*, 5(2):254–261, February 1996.
49. K. Yue, K.M. Fiebig, P.D. Thomas, H.S. Chan, E.I. Shakhinovich and K.A. Dill. A test of lattice protein folding algorithms. *National Academy of Science*, 92:325–329, 1995.
50. M.J. Zaki. *Protein Structure Prediction*. Humana Press, 2007. second edition.
51. Y. Zhang. I-TASSER server for protein 3D structure prediction. *BMC. Bioinformatics*, 9:40, 2008.
52. Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl. Acad. Sci. U.S.A.*, 102(4):1029–1034, January 2005.