2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

A Local Search Embedded Genetic Algorithm for Simplified Protein Structure Prediction

Mahmood A Rashid*†, M.A.Hakim Newton*, Md Tamjidul Hoque‡ and Abdul Sattar*†
*Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Australia
†Queensland Research Laboratory, National ICT Australia
‡Computer Science, University of New Orleans, USA
Email: {m.rashid,hakim.newton,a.sattar}@griffith.edu.au, thoque@uno.edu

***Abstract*—No single algorithm suits the best for the protein structure prediction problem. Therefore, researchers have tried hybrid techniques to mix the power of different strategies to gain improvements. In this paper, we present a hybrid search framework that embeds a tabu-based local search within a population based genetic algorithm. We applied our hybrid algorithm on simplified protein structure prediction problem. We use a low-resolution *ab initio* search method with the hydrophobic-polar energy model and face-centred-cubic lattice. Within the genetic algorithm, we apply local search in two different situations: *i)* only once at the beginning and *ii)* every time at search stagnation. At the beginning, we apply local search to improve the randomly generated individuals and use them as an initial population for the genetic algorithm. Later, we apply local search after applying a random-walk at situations where the genetic algorithm gets stuck. In both cases, the use of local search is to improve the randomised solutions quickly. We experimentally show that our hybrid approach outperforms the state-of-the-art approaches.**

***Keywords*—Hybrid Algorithm; Local Search; Genetic Algorithm; Protein Structure Prediction; HP Model; Lattice Model;**

I. Introduction

Proteins are essentially sequences of amino acids. They adopt specific folded three-dimensional structures to perform specific tasks. The function of a given protein is determined by its *native* structure, which has the lowest possible free energy level. Nevertheless, misfolded proteins cause many critical diseases such as Alzheimer's disease, Parkinson's disease, and Cancer [1], [2]. Protein structures are important in drug design and biotechnology.

Protein structure prediction (PSP) is computationally a very hard problem [3]. Given a protein's amino acid sequence, the problem is to find a three dimensional structure of the protein such that the total interaction energy amongst the amino acids in the sequence is minimised. The protein folding process that leads to such structures involves very complex molecular dynamics [4] and unknown energy factors. Researchers have used discretised lattice-based structures and simplified energy models [5]–[7] in an hierarchical approach for high resolution protein structure prediction. However, the complexity of the simplified problem still remains challenging.

There are a large number of existing search algorithms that attempt to solve the PSP problem by exploring feasible structures called *conformations*. For population based approaches, a genetic algorithm (GA^+) [8] reportedly produces the state-of-the-art results. However, for local search approaches, spiral search (SS-Tabu) [9], a tabu-based local search produces the best results.

In general, the success of both single-point search or population based search algorithms crucially depends on the balance of diversification and intensification of the exploration. However, these algorithms often get stuck or stall in local minima. As a result, they perform poorly on large sized ($length > 100$ amino acids) proteins. Any further progress to these algorithms require addressing the above issues appropriately.

In this paper, we present a hybrid search technique that embeds the *Spiral Search* algorithm (SS-Tabu) [9] within an enhanced population based *Genetic Algorithm* (GA^+) [8]. The random-walk [10] is a major addition in GA^+ to enhance the GA's performance particularly when it get stuck. In most GAs, after a number of generations, the individual conformations in the population become similar. The search often get stuck in this situation (called local minima). In [10], a random-walk algorithm is applied to break the stagnation. In random-walk, operators are used randomly to generate new solutions. However, after applying long random-walks, the quality of the individual solutions in the population drops significantly. Typically GAs take long time to regain from the dropped energy level but the local search can regain the energy level quickly. We apply local search in two different stages within GA^+: *i)* only once at the beginning and *ii)* every time at search stagnation. At the beginning of the search, we apply local search to improve the individuals that help GA^+ start with a rich initial population. We also apply local search at stagnation after applying a random-walk [10]. The random-walk algorithm diversifies the population widely. However, the SS-Tabu is applied following the random-walk to improve the diversified solutions quickly. In summary, the GA^+ is widening the search space and the SS-Tabu is deepening the search. We tested our hybrid algorithm on simplified protein structure prediction (PSP) problem. In our low-resolution *ab initio* method, we use hydrophobic-polar (HP) energy model for conformation evaluation and face-centred-cubic (FCC) lattice for structure mapping. We experimentally show that our hybrid algorithm preforms significantly better than the state-of-the-art approaches.

The rest of the paper is organized as follows: Section II illustrates the PSP problem and simplified models for PSP; Section III presents the related work; Section IV and V respectively present the GA^+ and the SS-Tabu framework used in our hybrid approach; Section VI describes our hybrid approach in detail; Section VII discusses and analyzes the experimental results; and finally, Section VIII presents our conclusions and outlines our future work.

978-1-4799-0454-9/13/$31.00 ©2013 IEEE

II. Background

Homology modeling, *protein threading* and *ab initio* are three computational approaches used in protein structure prediction. Prediction quality of *homology modeling* and *protein threading* depends on the sequential similarity of previously known protein structures. However, our work is based on the *ab initio* approach that only depends on the amino acid sequence of the target protein. Levinthal's paradox [11] and Anfinsen's hypothesis [12] are the basis of *ab initio* method for PSP. The idea was originated in 1970 when it was demonstrated that all information needed to fold a protein resides in its amino acid sequence. In our simplified protein structure prediction model, we use 3D FCC lattice for conformation mapping, HP energy model for conformation evaluation, and a hydrophobic-core centric local search algorithm (SS-Tabu) for conformation search. The simplified models, local search, and genetic algorithms are described below.

A. Simplified Model

To explore an astronomically large search space and to evaluate the conformations using a real energy model is a big challenge for existing search algorithms in PSP. In our approach, we use 3D FCC lattice points for conformation mapping and hydrophobic-polar (HP) energy model to keep the complexity manageable. The 3D FCC lattice and the HP energy model are briefly describe below.

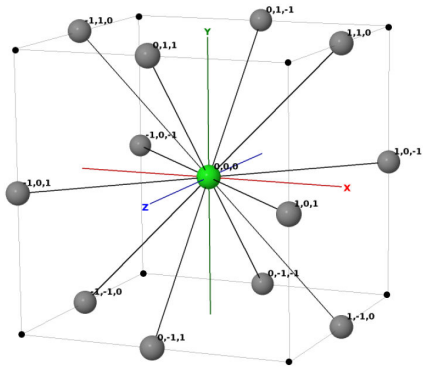


Figure 1: A unit 3D FCC lattice with 12 basis vectors on the Cartesian coordinates.

3D FCC Lattice: The FCC lattice has the highest packing density compared to the other existing lattices [13]. In FCC, each lattice point (the origin in Figure 1) has 12 neighbours with 12 *basis vectors* $(1,1,0)$, $(-1,-1,0)$, $(-1,1,0)$, $(1,-1,0)$, $(0,1,1)$, $(0,1,-1)$, $(1,0,1)$, $(1,0,-1)$, $(0,-1,1)$, $(-1,0,1)$, $(0,-1,-1)$, and $(-1,0,-1)$. The hexagonal closed pack (HCP) lattice, also known as cuboctahedron, was used in [14]. In HCP, each lattice point has 12 neighours that correspond to 12 basis vertices with real-numbered coordinates. The real numbers cause the loss of structural precision for PSP. In simplified PSP, conformations are mapped on the lattice by a sequence of basis vectors, or by the *relative vectors* that are relative to the previous basis vectors in the sequence.

HP Energy Model: The 20 constituent amino acids of proteins are broadly divided into two categories based on the hydrophobicity of the amino acids: (a) hydrophobic amino acids (*Gly, Ala, Pro, Val, Leu, Ile, Met, Phe, Tyr, Trp*) denoted as H; and

	H	P
H	-1	0
P	0	0

Figure 2: HP energy model [15]

(b) hydrophilic or polar amino acids (*Ser, Thr, Cys, Asn, Gln, Lys, His, Arg, Asp, Glu*) denoted as P. In the HP model [15], when two non-consecutive hydrophobic amino acids become topologically neighbours, they contribute a certain amount of negative energy, which for simplicity is shown as -1 in Figure 2. The total energy (E) of a conformation based on the HP model becomes the sum of the contributions of all pairs of non-consecutive hydrophobic amino acids as shown in Equation 1.

$$E = \sum_{i<j-1} c_{ij}.e_{ij} \tag{1}$$

Here, $c_{ij} = 1$ if amino acids i and j are non-consecutive neighbours on the lattice, otherwise 0; and $e_{ij} = -1$ if ith and jth amino acids are hydrophobic, otherwise 0.

Protein structures have hydrophobic cores (H-core) that hide the hydrophobic amino acids from water and expose the polar amino acids to the surface to be in contact with the surrounding water molecules [16]. H-core formation is the main objective of HP based PSP. To build H-cores, the total distance of all H-H pairs is minimised in [17]. A predefined motif based segment replacement strategy is applied in [14].

B. Local Search

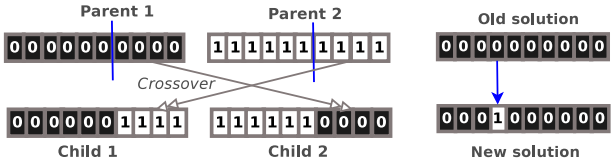
Starting from an initial solution, local search algorithms move from one solution to another to find a better solution. Local search algorithms are well known for efficiently producing high quality solutions, which are difficult for systematic search approaches. However, they are incomplete [18], and suffer from revisitation and stagnation. Restarting the whole or parts of a solution remains the typical approach to deal with such situations.

Tabu Meta-heuristic: Tabu meta-heuristic [19], [20] enhances the performance of local search algorithms. It maintains a short-term memory structure to remember the local changes of a solution. Then, any local changes for those stored positions are forbidden for certain number of subsequent iteration (known as tabu tenure).

C. Genetic Algorithms

GAs are a population-based search for optimisation problems. A genetic algorithm maintains a set of solutions known as population. In each *generation*, it generates a new population from the current population using a given set of genetic operators known as *crossover* and *mutation*. It then replaces inferior solutions by superior newly generated solutions to get a better current population. A typical crossover operator

randomly splits two solutions at a randomly selected crossover point and exchanges parts between them (Fig. 3a). A typical mutation operator alters a solution at a random point (Fig. 3b). In the case of PSP, conformations are regarded as solutions of a GA. Below we describe genetic operators used in PSP.



(a) Crossover (b) Mutation

Figure 3: Typical (a) crossover and (b) mutation operators

Crossover Operators: The crossover operators are applied on two selected parent conformations to exchange their parts to generate children conformations. In a *single-point crossover*, both parents are splitted at a single point (Fig. 4 a) while in a *multi-point crossover* they are splitted at more than one point. Nevertheless, the crossover operations succeed if they produce conformations that satisfy the self-avoiding walk constraint.

Mutation Operators: The mutation operators are applied on a single conformation. The operators can perform single-point change or multi-point changes. The mutation operations succeed if the resultant conformation remains a self-avoiding walk on the lattice.

III. Related Work

Different types of metaheuristic have been used in solving the simplified PSP problem. These include Monte Carlo Simulation [21], Simulated Annealing [22], Genetic Algorithms (GA) [23], [24], Tabu Search with GA [25], Tabu Search with Hill Climbing [26], Ant Colony Optimisation [27], Immune Algorithms [28], Tabu-based Stochastic Local Search [17], [29], and Constraint Programming [30]. Cebrian *et al.* [17] used tabu-based local search, and Shatabda *et al.* [29] used memory-based local search with tabu heuristic and achieved the state-of-the-art results. However, Dotu *et al.* [30] used constraint programming and found promising results but only for smaller sized ($length < 100$ amino acids) proteins. Besides local search, Unger and Moult [23] applied population based search algorithms (known as GA) to PSP and found their method to be more promising than the Monte Carlo based methods [21]. They used absolute encodings on the square and cubic lattices for HP energy model. Later, Patton [31] used relative encodings to represent conformations and a penalty method to enforce the self-avoiding walk constraint. The GA has been used by Hoque *et al.* [14] for cubic, and 3D HCP lattices. They used DFS-generated pathways [32] in GA crossover for protein structure prediction. They also introduced a twin-removal operator [33] to remove duplicates from the population to prevent the search from stalling.

No single algorithm suits the best for the protein structure prediction problem. Therefore, researchers have tried hybrid techniques to mix the power of different strategies to gain improvements. Ullah *et al.* in [34] and [35] combined local search with constraint programming. They used a 20×20 [36] energy model on FCC lattice and found promising results. In another hybrid approach [37], tabu meta-heuristic was combined with a genetic algorithm in two-dimensional HP model to observe crossover and mutation rate over time.

However, a new genetic algorithm *GA*$^+$ [8] and a tabu based local search algorithm *Spiral Search* [9] produce the current state-of-the-art results for HP energy model on 3D FCC lattice.

IV. GA$^+$: An Enhanced Genetic Algorithm

GA$^+$ [8] is an enhanced genetic algorithm for simplified PSP problem. It uses HP energy model and FCC lattice. The *pseudocode* of GA$^+$ is presented in Algorithm 1. It uses an exhaustive generation approach to diversify the search, a hydrophobic core-directed macro move to intensify the search, and a random-walk algorithm to recover from stagnation.

Algorithm 1: gaPlus(opR,rwT)

```
1  op: Operators, c, c': Conformations
2  opR: Operator selection probabilities
3  curP,newP: Current and new populations
4  rwT: Number of non-improving
5       generations before random walk.
6  //====================
7  initPopulation(curP)
8  foreach Generation until timeout do
9      selectOperator(op, opR)
10     if mutation(op) then
11         foreach c ∈ curP do
12             newP.add(mutConf(c))
13     else //crossover(op)
14         while ¬full(newP) do
15             c, c' ← randomConfs(curP)
16             newP.add(crsConfs(c, c'))
17     if ¬improved(newP, rwT) then
18         randomWalk(newP)
19     curP ← newP
20 return bestConformation(curP)
```

A. Exhaustive Generation

Unlike traditional GA, in GA$^+$, the use of randomness is reduced significantly by an exhaustive generation approach. For mutation operators, GA$^+$ adds one resultant conformation for *each* conformation in the current population to the new population. Operators are applied to all possible points exhaustively until finding a better solution than the parent. If no better solution is found, the parent survives through the next generation. On the other hand, for crossover operators, two resultant conformations are added to the new population from two randomly selected parent conformations. Crossover operators generate child conformations by applying the crossover operator in all possible points on two randomly selected parents. The best two conformations from the parents and the children are then become the resultant conformations for the next generation.

B. Macro-move

Macro-move is a composite operator (Figure 5) that uses a series of diagonal-moves (Figure 4 c) on a given conformation to build the H-core around the hydrophobic-core-center (HCC). The macro-move squeezes the conformation and quickly forms

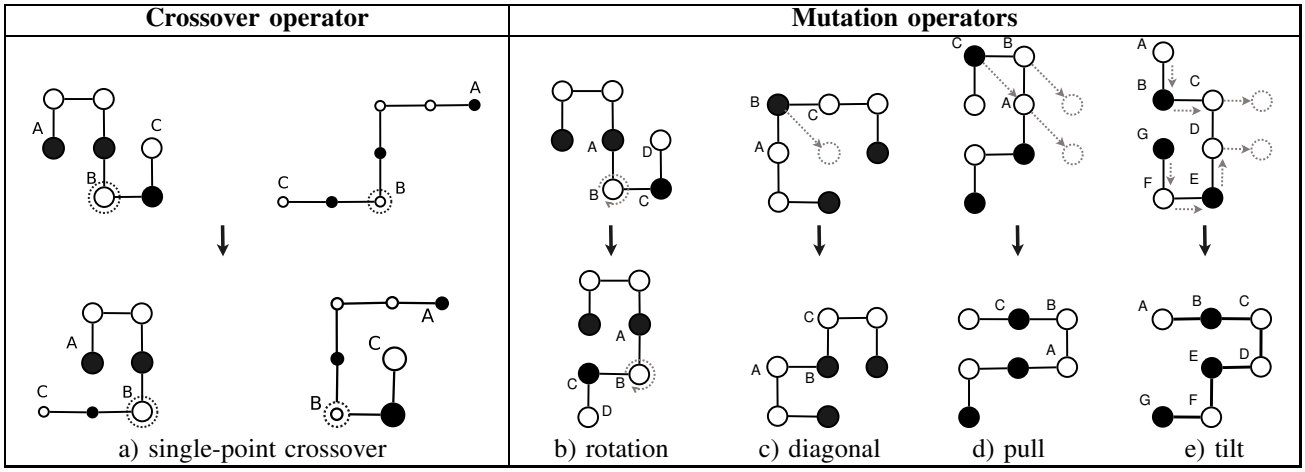


Figure 4: The operators that are used in our GA$^+$ on 3D FCC lattice space. For simplification and easy understanding the figures are presented in 2D space. The black solid circles represent the hydrophobic amino acids and others are hydrophilic.

the H-core. In GA$^+$, macro-move is used as a mutation operator.

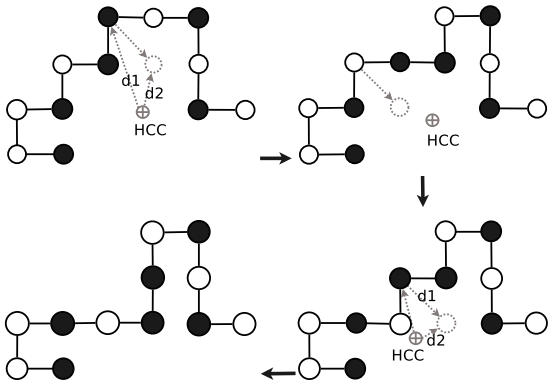


Figure 5: A macro move operator comprising a series of diagonal moves. For simplification and easy understanding the figures are presented in 2D space.

In macro-move, the HCC is calculated by finding arithmetic means of x, y, and z coordinates of all hydrophobic amino acids. The macro-move for a given number of iterations repeatedly applies the diagonal move either at each P- or at each H-type amino acid positions. Whether to apply the diagonal move on P- or H-type amino acids is determined by using a Bernoulii distribution with probability p (typically $p = 20\%$ for P-type amino acids). For a P-type amino acid, the first successful diagonal move is considered. Whereas, for a H-type amino acid, the first successful diagonal move that does not increase the Cartesian distance of the amino acid from the HCC is taken. All the amino acids are traversed and the successful moves are applied as one composite move.

C. Stagnation Recovery

In GA$^+$, when the search stuck in a local minima, a random-walk algorithm is applied to recover from stagnation. This algorithm helps break the pre-matured H-cores of the individuals. To restart, the GA$^+$ accepts those conformations that are close to the respective parent conformations in terms of energy level, and have possible maximum structural diversity from the respective parent conformations. For genetic algorithms, random-walk is very effective [10] to recover from stagnation.

D. The Primitive GA Operators

Along with exhaustiveness, macro-move and random-walk, the primitive operators (as shown in Fig 4) that are implemented in GA$^+$ are single-point crossover (Figure 4 a), rotation mutation (Figure 4 b), diagonal move (Figure 4 c), pull moves (Figure 4 d), and tilt moves (Figure 4 e)

V. SS-Tabu: The Spiral Search

Spiral search (SS-Tabu) [9] is a local search guided by *tabu meta-heuristics*. In SS-Tabu, the diagonal move operator (as shown in Figure 4 c) is used in building H-core in a spiral fashion. The move is just a corner-flip to an unoccupied lattice point. A tabu list is maintained to control the amino acids to involve in diagonal move. The *pseudocode* of SS-Tabu, as shown in Algorithm 2, is composed of moves selection (Algorithm 2: Line 4 and 9) and local minima handling sub-procedures (Algorithm 2: Line 20 and 24).

A. Move Selection

In move selection, the H amino acids get priority in comparison to P amino acids. The H-move selection is guided by the Cartesian distance between HCC and the H amino acids in the sequence. For the ith hydrophobic amino acid, the common topological neighbours (TN) of the $(i-1)$th and $(i+1)$th amino acids are computed. The TN of a lattice point are the points at unit lattice-distance apart from it. The Cartesian distance of all unoccupied common neighbours are calculated from the HCC. Then the point with the shortest distance is listed in the possible H-move list for ith hydrophobic amino acid if its current distance from HCC is greater than that of the selected point. When all H amino acids are traversed and the feasible shortest distances are listed in H-move list, the amino acid having the shortest distance in H-move list is chosen to apply a diagonal move. A *tabu list* is maintained for each hydrophobic amino acid to control the selection priority amongst them. For each successful move, the *tabu list* is updated for the respective H amino acid. For P amino acids,

the same diagonal moves are applied as H-move. However, no hydrophobic-core-center is calculated, no Cartesian distance is measured, and no *tabu list* is maintained for P-move.

Algorithm 2: *SSTabu*(**maxIter, maxRetry, maxRW, c**)

```
1  //H and P are hydrophobic & polar amino acids.
2  initTabuList()
3  for (i = 1 to maxIter) do
4      mv ⟵ selectMoveForH()
5      if (mv != null) then
6          applyMove(mv)
7          updateTabuList(i)
8      else
9          mv ⟵ selectMoveForP()
10         if (mv != null) then
11             applyMove(mv)
12     evalute(AA)  //AA-amino acid array
13     if (!improved) then
14         retry++
15     else
16         improvedList ⟵ addTopOfList()
17         retry = 0
18         rw = 0
19     if retry ≥ maxRetry then
20         randomWalk(maxPull)
21         resetTabuList()
22         rw++;
23     if rw ≥ maxRW then
24         relayRestart(improvedList)
25         resetTabuList()
```

B. Stagnation Recovery

For hard optimisation problems such as protein structure prediction, local search algorithms often face stagnation. Thus, handling such situation intelligently is important to proceed further. In SS-Tabu, random-walk [10] and relay-restart technique are applied on an on-demand basis to deal with stagnation. The random-walk algorithm is applied to break the pre-matured H-cores. For local search, the random-walk is found effective [10] in stagnation recovery. Relay-restart is applied when random-walk fails to escape from local minima. In relay-restart, instead of using a fresh restart or restarting from the current best solution, search restarts from an improving solution. An improving solution list is maintained that contains all the improving solutions after initialisation.

VI. Our Approach: Local Search Embedded GA

In this paper, we present a hybrid search algorithm that embeds a *SS-Tabu* within a population based *GA*$^+$. We tested the algorithm on simplified protein structure prediction problem. In our low-resolution *ab initio* method, we use HP energy model for conformation evaluation and FCC lattice for structure mapping. The hybrid search framework and its components are presented below:

A. Hybrid Framework:

The hybrid framework (as shown in Algorithm 3) is the combination of the GA$^+$ [8] and the spiral search (SS-Tabu) [9]. We apply SS-Tabu in two different stages within GA$^+$ as follows:

Algorithm 3: *LSEmbeddedGA*(**opR, rwT**)

```
1  op: Operators, c, c': Conformations
2  opR: Operator selection probabilities
3  curP,newP: Current and new populations
4  rwT: Number of non-improving
5       generations before random walk.
6  //====================
7  initPopulation(curP)
8  SSTabu(curP)
9  foreach Generation until timeout do
10     selectOperator(op, opR)
11     if mutation(op) then
12         foreach c ∈ curP do
13             newP.add(mutConf(c))
14     else //crossover(op)
15         while ¬full(newP) do
16             c, c' ← randomConfs(curP)
17             newP.add(crsConfs(c, c'))
18     if ¬improved(newP, rwT) then
19         randomWalk(newP)
20         SSTabu(newP)
21     curP ← newP
22 return bestConformation(curP)
```

1) At the beginning: In GA$^+$, a randomly generated population was the starting point of the genetic operations. In our hybrid approach, we apply a H-core directed local search guided by tabu meta-heuristic (Algorithm 3 line 8) to improve all the randomly generated individuals within the population. These improved individuals are then used as the initial population of the hybrid GA$^+$. In this stage the population based genetic algorithm gain a lift at the starting.

2) At the stagnation: In most cases, after a number of generations, the individual conformations in the GA population become similar due to the formation of pre-matured H-cores. In this situation (called a local minimum), the search often get stuck. In [10], a random-walk algorithm is applied to escape the stagnation by breaking the pre-matured H-cores. In a random-walk, both bad and good solutions are generated by using pull moves. However, after applying long random-walks, the quality of the individual solutions in the population normally drops significantly. Typical GAs take long time to regain from the dropped energy level. However, local search algorithms can regain the energy level very quickly. Moreover, during the application of pull moves, we observe energy level and structural diversification of the generated structures and maintain a balance between these two. We allow energy level to change within 5% to 10% and the structure within 10% to 75%. We try to accept the conformation that is close to the current conformation in terms of energy level and has possible maximum structural diversity from the current conformation. In this process, the H-cores of the individuals are broken and eventually the fitness of the individuals in terms of free energy level drops. At this point, immediately after applying random-walk (Algorithm 3 line 20), we apply SS-Tabu to improve the individuals in the population quickly.

B. Further Implementation Details

Like other search algorithms, our hybrid search requires initialisation. It also needs evaluation of the solution in each iteration. An initial population of individual solutions are generated and enhanced by local search before starting genetic algorithms.

Algorithm 4: *initialise*()

```
1  //AA-amino acid array of the protein
2  //SAW- Self-avoiding-walk
3  basisVec[12] ⟵ getTwelveBasisVectors()
4  AA[0] ⟵ AminoAcid(0, 0, 0)
5  while (!SAW) do
6      for (i = 1 to seqLength − 1) do
7          k ⟵ getRandom(12)
8          basis ⟵ basisVec[k]
9          node ⟵ AA[i − 1] + basis
10         if isFree(node) then
11             AA[i] ⟵ AminoAcid(node)
12         else
13             SAW ⟵ false
14             break
15 return AA[ ]
```

1) Initialisation: Our algorithm starts with a feasible set of conformation known as population. We generate an initial conformation following a self-avoiding walk (SAW) on FCC lattice points. The *pseudocode* of the algorithm is presented in Algorithm 4. It places the first amino acid at $(0, 0, 0)$. It then randomly selects a basis vector to place the successive amino acid at a neighbouring free lattice point. The mapping proceeds until a self avoiding walk is found for the whole protein sequence.

2) Evaluation: After each iteration, the conformation is evaluated by counting the H-H contacts (topological neighbour) where the two amino acids are non-consecutive. The *pseudocode* in Algorithm 5 presents the algorithm of calculating the free energy of a given conformation. Note that the energy value is negation of the of the H-H contact count.

Algorithm 5: *evaluate*(**AA**)

```
1  for (i = 1 to seqLength − 1) do
2      for (k = i + 2 to seqLength − 1) do
3          if AAType[i] = AAType[k] = H then
4              nodeI ⟵ AA[i]
5              nodeJ ⟵ AA[k]
6              sqrD ⟵ getSqrDist(nodeI, nodeJ)
7              if sqrD = 2 then
8                  fitness ⟵ fitness − 1
9  return fitness
```

VII. Experimental Results and Analysis

In our experiment, the protein instances (as shown in Table I), the *S*, *F180*, and *R* instances are taken from Peter Clote laboratory website[1]. These instances have been used in [8], [9], [17], [29], [30] for evaluating different algorithms. We also use five larger sequences that are taken from the CASP[2] competition. The corresponding CASP target IDs for proteins *3mse*, *3mr7*, *3no6*, *3no3*, and *3on7* are *T0521*, *T0520*, *T0516*, *T0570*, and *T0563*. These CASP targets are also used in [29]. To fit in the HP model, the CASP targets are converted to HP sequences based on the hydrophobic properties of the constituent amino acids. The lower bounds of the free energy values (in Column *LBFE* of Table I) are obtained from [17], [29]; however, there are some unknown values (presented as **n/a**) of lower bounds of free energy for large sequences.

A. Data Table (Table I)

In Table I, we present three different sets of result obtained from *i*) Local Search [9] (SS-Tabu), *ii*) Genetic Algorithms [8] (GA$^+$), and *iii*) Hybrid Algorithms (Hybrid-GA). We compare our reseults with [8] and [9] because these two algorithms produce the current state-of-the-art results for the same models. In the table, the *Size* column presents the number of amino acids in the sequences, and the *LBFE* column shows the known lower bounds of free energy for the corresponding protein sequences in Column *ID*. However, a lower bound of free energy for protein *3on7* is unknown. The best and average free energy for three different algorithms are also present in the table. The **bold-faced** values indicate better performance in comparison to the other algorithms for corresponding proteins. The experimental results show that our Hybrid-GA wins over SS-Tabu and GA$^+$ over the 15 proteins with a significant margin on average search results.

B. Relative Improvement

The difficulty to improve energy level is increased as the predicted energy level approaches to the lower bound. For example, if the lower bound of free energy of a protein is -100, the efforts to improve energy level from -80 to -85 is much less than that to improve energy level from -95 to -100 though the change in energy is the same (-5). Relative Improvement (RI) explains how close our predicted results to the lower bound of free energy with respect to the energy obtained from the state-of-the-art approaches.

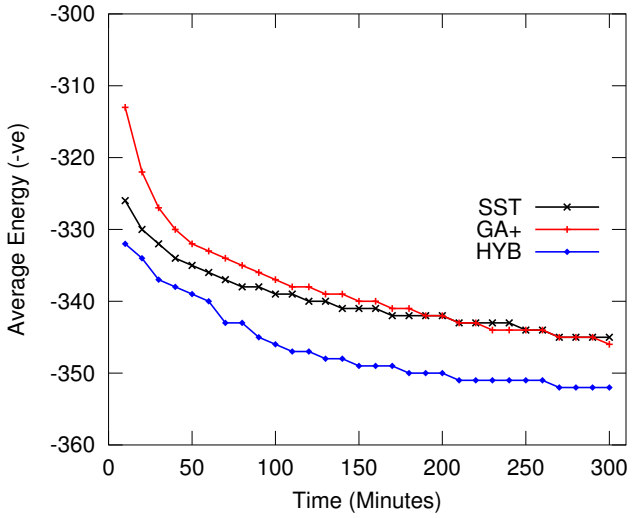


Figure 6: Search progress for protein R1 with time. SST, GA$^+$, and HYB are spiral search, genetic algorithms and new hybrid algorithm.

[1]Peter Clote lab: bioinformatics.bc.edu/clotelab/FCCproteinStructure
[2]CASP9: predictioncenter.org/casp9/targetlist.cgi

Protein Info			Hybrid-GA (GA$^+$ & SS-Tabu)		The current state-of-the-art results						Time
					Spiral Search [9] (SS-Tabu)			Genetic Algorithm [8] (GA$^+$)			
ID	**Size**	**LBFE**	Best	**Avg**	**Best**	**Avg**	**RI**	**Best**	**Avg**	**RI**	**mins**
S1	135	-357	**-353**	**-349**	-355	-347	20%	-355	-348	11%	
S2	151	-360	**-355**	**-352**	-354	-347	**38%**	-356	-349	27%	120
S3	162	-367	**-360**	**-355**	-359	-350	29%	-361	-349	**33%**	
S4	164	-370	**-363**	**-356**	-358	-350	30%	-364	-352	22%	
F180_1		-378	**-359**	**-348**	-357	-340	22%	-351	-341	18%	
F180_2	180	-381	**-365**	**-353**	-359	-345	22%	-362	-346	19%	300
F180_3		-378	**-371**	**-359**	-362	-353	25%	-361	-350	32%	
R1		-384	**-364**	**-352**	-359	-345	18%	-355	-346	17%	
R2	200	-383	**-364**	**-355**	-358	-346	23%	-360	-346	23%	300
R3		-385	**-366**	**-353**	-365	-345	19%	-363	-344	22%	
3mse	179	-323	**-293**	**-286**	-289	-280	14%	-290	-279	17%	
3mr7	189	-355	**-331**	**-320**	-328	-313	17%	-328	-316	10%	
3no6	229	-455	**-424**	**-406**	-411	-391	23%	-420	-400	12%	300
3no3	258	-494	**-426**	**-407**	-412	-393	**14%**	-421	-402	**6%**	
3on7	279	n/a	**-526**	**-501**	-512	-485	n/a	-515	-485	n/a	

Table I: Experimental results of new hybrid approach, SS-Tabu, and GA$^+$. Columns *RI* present the relative improvements over the state-of-the-art approaches. The results are obtained from 50 different runs of similar setting for each protein.

$$\mathsf{RI} = \frac{E_t - E_r}{E_l - E_r} * 100\% \quad (2)$$

In Table I, we also present a comparison of improvements (%) on average conformation quality (in terms of free energy levels). We compare Hybrid-GA (target) with SS-Tabu and GA$^+$ (references). For each protein, the RI of the target (t) w.r.t. the reference (r) is calculated using the formula in Equation 2, where E_t and E_r denote the average energy values achieved by the target and the reference respectively, and E_l is the lower bound of free energy for the protein in the HP model. We present the relative improvements only for the proteins having known lower bound of free energy values. We test our new algorithm on 15 different proteins of various length. The **bold-faced** values are the minimum and the maximum improvements for the same column.

Improvement w.r.t. SS-Tabu: The experimental results in Table I, at column *RI* under SS-Tabu shows that our Hybrid-GA is able to improve the search quality in terms of minimising the free energy level over all the 15 proteins considered for the test. The relative improvements with respect to SS-Tabu range from 14% to 38%.

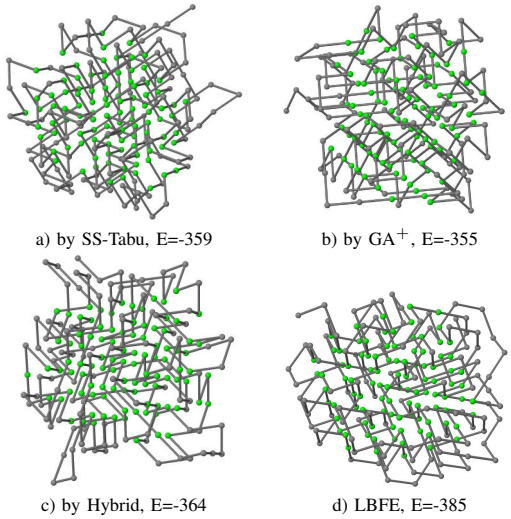
Improvement w.r.t. GA$^+$: The experimental results in Table I, at column *RI* (relative improvement) under GA$^+$ shows that our Hybrid-GA is able to improve the search quality in terms of minimising the free energy level over all 15 proteins considered for the test. The relative improvements with respect to GA$^+$ range from 6% to 33%.

C. Search progress

We compare the search progresses of different approaches; SS-Tabu, GA$^+$, and Hybrid-GA over time. Figure 6 shows the average energy values obtained with times by the algorithms for protein R1 over 50 different runs. We observe that all of the algorithms achieve very good progress initially, but with time increasing, our Hybrid-GA makes more progress than SS-Tabu and GA$^+$. Notice that the Hybrid-GA curve does not coincide with the SS-Tabu curve although the former uses the latter to initialise its population. This is because SS-Tabu continues the search only with one solution all the time while Hybrid-GA runs SS-Tabu on each individual (30 in number) in the population for a short time (2 minutes). This is similar to 30 fresh restarts in first 60 minutes and we take the best result obtained so far at each time point. The Hybrid-GA curve is thus different.

D. Simplified structure

In Figure 7, we show the best structures found by Hybrid-GA, SS-Tabu and GA$^+$ for protein $R1$. Each algorithm is run over a period of 5 hours to achieve the results. However, the structure in Figure 7-d is collected from the literature [17], [29]. To view structures, we use *Jmol*[3]: an open-source Java viewer for chemical structures in 3D.



a) by SS-Tabu, E=-359 b) by GA$^+$, E=-355

c) by Hybrid, E=-364 d) LBFE, E=-385

Figure 7: 3D structures of protein R1 obtained by different approaches.

[3]Jmol website: www.jmol.org

VIII. Conclusion

In this paper, we presented a hybrid genetic algorithm that integrated a tabu-based hydrophobic-core directed local search within a genetic algorithm framework. In our low-resolution *ab initio* method, we use hydrophobic-polar energy model and face-centred-cubic lattice for protein structure prediction. We apply local search (*i*) once at the beginning: to build a rich initial population for genetic algorithm and (*ii*) later every time at stagnation: to improve the diversified individuals after applying random-walk. We experimentally show that our hybrid approach outperforms the state-of-the-art approaches. In future, we intend to apply our hybrid algorithm in high resolution protein structure prediction.

Acknowledgment

We would like to express our great appreciation to the people managing the *Cluster Computing Services* at National ICT Australia (NICTA) and Griffith university. They helped a lot in preparing this article on time by taking care of our submitted jobs in clusters.

References

[1] Adam Smith, "Protein misfolding," *Nature Reviews Drug Discovery*, vol. 426, no. 6968, pp. 78–102, December 2003.

[2] C. M. Dobso, "Protein folding and misfolding," *Nature*, vol. 426, no. 6968, pp. 884–890, 2003.

[3] The Science Editorial, "So much more to know," *The Science*, vol. 309, no. 5731, pp. 78–102, July 2005.

[4] R. Bonneau and D. Baker, "*Ab initio* protein structure prediction: progress and prospects," *Annual Review of Biophysics and Biomolecular Structure*, vol. 30, no. 1, pp. 173–89, 2001.

[5] C. Rohl, C. Strauss, K. Misura, and D. Baker, "Protein structure prediction using Rosetta," *Methods in enzymology*, vol. 383, pp. 66–93, 2004.

[6] J. Lee, S. Wu, and Y. Zhang, "Ab initio protein structure prediction," *From protein structure to function with bioinformatics*, pp. 3–25, 2009.

[7] Y. Xia, E. S. Huang, M. Levitt, and R. Samudrala, "*Ab initio* construction of protein tertiary structures using a hierarchical approach," *Journal of Mol. Biology*, 2008.

[8] M. A. Rashid, M. Hoque, M. A. H. Newton, D. Pham, and A. Sattar, "A new genetic algorithm for simplified protein structure prediction," in *AI 2012: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, 2012.

[9] M. A. Rashid, M. A. H. Newton, M. T. Hoque, S. Shatabda, D. Pham, and A. Sattar, "Spiral search: a hydrophobic-core directed local search for simplified PSP on 3D FCC lattice," *BMC Bioinformatics*, vol. 14, no. Suppl 2, p. S16, 2013.

[10] M. A. Rashid, S. Shatabda, M. A. H. Newton, M. T. Hoque, D. N. Pham, and A. Sattar, "Random-walk: a stagnation recovery technique for simplified protein structure prediction," in *ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, 2012, pp. 620–622.

[11] C. Levinthal, "Are there pathways for protein folding?" *Journal of Medical Physics*, vol. 65, no. 1, pp. 44–45, 1968.

[12] C. B. Anfinsen, "The principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[13] T. Hales, "A proof of the Kepler conjecture," *The Annals of Mathematics*, vol. 162, no. 3, pp. 1065–1185, 2005.

[14] M. T. Hoque, M. Chetty, and A. Sattar, "Protein folding prediction in 3D FCC HP lattice model using genetic algorithm," vol. 2007. IEEE Congress on Evolutionary Computation, 2007, pp. 4138–4145.

[15] K. F. Lau and K. A. Dill, "A lattice statistical mechanics model of the conformational and sequence spaces of proteins," *Macromolecules*, vol. 22, no. 10, pp. 3986–3997, 1989.

[16] K. Yue and K. A. Dill, "Sequence-structure relationships in proteins and copolymers," *Physical Review E*, vol. 48, no. 3, p. 2267, 1993.

[17] M. Cebrián, I. Dotú, P. Van Hentenryck, and P. Clote, "Protein structure prediction on the face centered cubic lattice by local search," in *Proceedings of the 23rd national conference on Artificial intelligence - Volume 1*, 2008, pp. 241–246.

[18] B. Berger and T. Leightont, "Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete," *Journal of Computational Biology*, vol. 5, no. 1, pp. 27–40, 1998.

[19] F. Glover and M. Laguna, *Tabu search*. Kluwer Academic Pub, 1998, vol. 1.

[20] F. Glover, "Tabu search - part I ," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[21] C. Thachuk, A. Shmygelska, and H. H. Hoos, "A replica exchange Monte Carlo algorithm for protein folding in the HP model," *BMC bioinformatics*, vol. 8, no. 1, p. 342, 2007.

[22] A.-A. Tantar, N. Melab, and E.-G. Talbi, "A grid-based genetic algorithm combined with an adaptive simulated annealing for protein structure prediction," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 2008.

[23] R. Unger and J. Moult, "A genetic algorithm for 3D protein folding simulations," Morgan Kaufmann Publishers. The 5th International Conference on Genetic Algorithms, 1993, p. 581.

[24] M. T. Hoque, "Genetic Algorithm for *Ab initio* Protein Structure Prediction based on Low Resolution Models," Ph.D. dissertation, Gippsland School of Information Technology, Monash University, Australia, Sep. 2007.

[25] H.-J. Böckenhauer, A. Z. M. D. Ullah, L. Kapsokalivas, and K. Steinhöfel, "A Local move set for protein folding in triangular lattice models," in *WABI*, ser. Lecture Notes in Computer Science, vol. 5251. Springer, 2008, pp. 369–381.

[26] G. W. Klau, N. Lesh, J. Marks, and M. Mitzenmacher, "Human-Guided Tabu Search," in *The Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 2002.

[27] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life reviews*, vol. 2, no. 4, pp. 353–373, 2005.

[28] V. Cutello, G. Nicosia, M. Pavone, and J. Timmis, "An immune algorithm for protein structure prediction on lattice models," *IEEE Trans on Evolutionary Computing*, vol. 11-1, pp. 101–117, 2007.

[29] S. Shatabda, M. A. H. Newton, D. N. Pham, and A. Sattar, "Memory-based local search for simplified protein structure prediction," in *ACM Conference on Bioinformatics, Computational Biology and Biomedicine*. ACM, 2012, pp. 345–352.

[30] I. Dotu, M. Cebrián, P. Van Hentenryck, and P. Clote, "On lattice protein structure prediction revisited," *IEEE Transactions on Comp. Bio. and Bioinformatics*, 2011.

[31] A. L. Patton, W. F. Punch III, and E. D. Goodman, "A standard GA approach to native protein conformation prediction." Int. Conf. on Genetic Algorithms, 1995.

[32] M. T. Hoque, M. Chetty, A. Lewis, A. Sattar, and V. M. Avery, "DFS-generated pathways in GA crossover for protein structure prediction," *Neurocomputing*, vol. 73, no. 13-15, pp. 2308–2316, 2010.

[33] M. T. Hoque, M. Chetty, A. Lewis, and A. Sattar, "Twin removal in genetic algorithms for protein structure prediction using low-resolution model," *Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 1, pp. 234–245, 2011.

[34] A. D. Ullah, L. Kapsokalivas, M. Mann, and K. Steinhöfel, "Protein folding simulation by two-stage optimization," in *Computational Intelligence and Intelligent Systems*, ser. Communications in Computer and Information Science.

[35] A. D. Ullah and K. Steinhöfel, "A hybrid approach to protein folding problem integrating constraint programming with local search," *BMC bioinformatics*, vol. 11, no. Suppl 1, p. S39, 2010.

[36] M. Berrera, H. Molinari, and F. Fogolari, "Amino acid empirical contact energy definitions for fold recognition in the space of contact maps," *BMC bioinformatics*, vol. 4, no. 1, p. 8, 2003.

[37] T. Jiang, Q. Cui, G. Shi, and S. Ma, "Protein folding simulations of the hydrophobic–hydrophilic model by combining tabu search with genetic algorithms," *The Journal of chemical physics*, vol. 119, p. 4592, 2003.