Computational Biology and Chemistry 61 (2016) 162–177

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem

Research Article

Guided macro-mutation in a graded energy based genetic algorithm for protein structure prediction

Mahmood A. Rashid[a,d,*], Sumaiya Iqbal[b], Firas Khatib[c], Md Tamjidul Hoque[b], Abdul Sattar[d]

[a] SCIMS, University of the South Pacific, Laucala Bay, Suva, Fiji
[b] CS, University of New Orleans, LA, USA
[c] CIS, University of Massachusetts Dartmouth, MA, USA
[d] IIIS, Griffith University, Brisbane, QLD, Australia

ARTICLE INFO

Article history:
Received 23 May 2015
Received in revised form
29 November 2015
Accepted 21 January 2016
Available online 15 February 2016

Keywords:
Ab initio protein structure prediction
Genetic algorithms
FCC lattice
Miyazawa–Jernigan model
Hydrophobic-polar model

ABSTRACT

Protein structure prediction is considered as one of the most challenging and computationally intractable combinatorial problem. Thus, the efficient modeling of convoluted search space, the clever use of energy functions, and more importantly, the use of effective sampling algorithms become crucial to address this problem. For protein structure modeling, an off-lattice model provides limited scopes to exercise and evaluate the algorithmic developments due to its astronomically large set of data-points. In contrast, an on-lattice model widens the scopes and permits studying the relatively larger proteins because of its finite set of data-points. In this work, we took the full advantage of an on-lattice model by using a face-centered-cube lattice that has the highest packing density with the maximum degree of freedom. We proposed a graded energy—strategically mixes the Miyazawa–Jernigan (MJ) energy with the hydrophobic-polar (HP) energy—based genetic algorithm (GA) for conformational search. In our application, we introduced a 2×2 HP energy guided macro-mutation operator within the GA to explore the best possible local changes exhaustively. Conversely, the 20×20 MJ energy model—the ultimate objective function of our GA that needs to be minimized—considers the impacts amongst the 20 different amino acids and allow searching the globally acceptable conformations. On a set of benchmark proteins, our proposed approach outperformed state-of-the-art approaches in terms of the free energy levels and the root-mean-square deviations.

1. Introduction

Protein folding, by which the primary protein chain with amino acid residue sequence folds into its characteristics and functional three-dimensional (3D) structure in nature, is yet a very complex physical process to simulate (Morowitz, 1968; Stouthamer, 1973; Alberts et al., 2002). Once the folded 3D shape is available, it enables protein to perform specific tasks for living organisms. Conversely, misfolded proteins are responsible for various fatal diseases, such as prion disease, Alzheimer's disease, Huntington's disease, Parkinson's disease, diabetes, and cancer (Smith, 2003; Dobson, 2003). Because of these, protein structure prediction (PSP) problem has emerged as a very important research problem.

Homology modeling, threading and *ab initio* are the broad categories of available computational approaches. However, while homologous template is not available, *ab initio* becomes the only computation approach, which aims to find the three dimensional structure of a protein from its primary amino acid sequence alone such that the total interaction energy among the amino acids is minimized.

Ab initio computational approach for PSP is a daunting task (Dodson, 2007) and for modeling the structure on a realistic continuum space such as off-lattice space is even more daunting. However, there are several existing off-lattice models such as Rosetta (Kaufmann et al., 2010), Quark (Xu and Zhang, 2012), I-TASSER (Lee et al., 2009), and so on which map the structures on the realistic continuum spaces rather than using discretized on-lattice spaces and hence, those approaches need to deal with the astronomical data-points incurring heavy computational cost. On-lattice model on the other hand, (i) due to reduced complexity helps fast algorithms developments and (ii) widens the scope as well as permits relatively longer protein chains to examine, which is otherwise

* Corresponding author at: SCIMS, University of the South Pacific, Laucala Bay, Suva, Fiji.
E-mail addresses: mahmood.rashid@usp.ac.fj (M.A. Rashid), siqbal1@uno.edu (S. Iqbal), fkhatib@umassd.edu (F. Khatib), thoque@uno.edu (M.T. Hoque), a.sattar@griffith.edu.au (A. Sattar).

http://dx.doi.org/10.1016/j.compbiolchem.2016.01.008
1476-9271/

prohibitive (Miyazawa and Jernigan, 1985; Berrera et al., 2003; Lau and Dill, 1989; Cooper et al., 2010; Das and Baker, 2008; Wroe et al., 2005). The computed on-lattice fold can be translated to off-lattice space via hierarchical approaches to provide output in real-space (Hoque et al., 2005, 2010, 2011; Iqbal et al., 2015). The Monte Carlo (MC) or, Conformational Space Annealing (CSA) used in Rosetta can be replaced with better algorithm developed using on-lattice models (Hoque et al., 2005, 2010, 2011). For instance, we embedded one of our previous on-lattice algorithm (Hoque et al., 2011) within Rosetta and the embedded algorithm improved (Higgs et al., 2012a) the average RMSD by 9.5% and average TM-Score by 17.36% over the core Rosetta (Kaufmann et al., 2010). Similarly, the embedded algorithm also outperformed (Higgs et al., 2012a) I-TASSER (Lee et al., 2009). These improvements motivated us further developing superior algorithms using on-lattice models.

The two most important building blocks of an *ab initio* PSP are (i) an accurate (computable) energy function (Iqbal et al., 2015) and (ii) an effective search or sampling algorithm. For a simplified model based PSP, it is possible to compute the lower bound (Giaquinta and Pozzi, 2013). It is also possible to know what would be the best score and hence the native score of a sequence by exhaustive enumeration (Unger and Moult, 1993a; Lesh et al., 2003) (which is feasible to compute for smaller sequences only). Even though, there exists no efficient sampling algorithm yet that can conveniently obtain the known final structure starting from a random structure for all possible available cases (Hoque et al., 2005, 2007a). Therefore, a number of efforts are being made, such as, different types of meta-heuristics have been used in solving the on lattice PSP problems. These include Monte Carlo Simulation (Thachuk et al., 2007), Simulated Annealing (Tantar et al., 2008), Genetic Algorithms (GA) (Hoque et al., 2005, 2007a; Unger and Moult, 1993b; Hoque, 2008), Tabu Search with GA (Böckenhauer et al., 2008), GA with twin-removal operator (Hoque et al., 2011), Tabu Search with Hill Climbing (Klau et al., 2002), Ant Colony Optimization (Blum, 2005), Particle Swarm Optimization (Kondov and Berlich, 2011; Mansour et al., 2012), Immune Algorithms (Cutello et al., 2007), Tabu-based Stochastic Local Search (Cebrián et al., 2008; Shatabda et al., 2012), Firefly Algorithm (Maher et al., 2014), and Constraint Programming (Mann et al., 2008; Dotú et al., 2011).

Krasnogor et al. (2002) applied HP model for PSP problem using the square, triangular, and diamond lattices and further extended their work applying fuzzy-logic (Pelta and Krasnogor, 2005). Islam et al. further improved the performance of memetic algorithms in a series of work (Islam, 2011; Islam and Chetty, 2009; Islam et al., 2011c,a) for the simplified PSP models. They also proposed a clustered architecture for the memetic algorithm with a scalable niching technique (Islam and Chetty, 2010, 2013; Islam et al., 2011b) for PSP. However, using 3D FCC lattice points, the recent state-of-the-art results for the HP energy model have been achieved by genetic algorithms (Rashid et al., 2012a; Shatabda et al., 2013b), local search approaches (Shatabda et al., 2012; Rashid et al., 2013c), a local search embedded GA (Rashid et al., 2013a), and a multi-point parallel local search approach (Rashid et al., 2013b). Kern and Liao (2013) applied hydrophobic-core guided genetic operator for efficient searching on HP, HPNX and hHPNX lattice models. Several approaches towards the 20×20 energy model include a constraint programming technique used in Dal Palù et al. (2004, 2005) by to predict tertiary structures of real proteins using secondary structure information, a fragment assembly method (Dal Palù et al., 2011) to optimize protein structures. Among other successful approaches, a population based local search (Kapsokalivas et al., 2009) and a population based genetic algorithm (Torres et al., 2007) are found in the literature that applied empirical energy functions.

In a hybrid approach, Ullah and Steinhöfel (2010) applied a constraint programming based large neighborhood search technique on top of the output of COLA (Dal Palù et al., 2007) solver. The hybrid approach produced the state-of-the-art results for several small sized (less than 75 amino acids) benchmark proteins. In another work, Ullah et al. (2009) proposed a two stage optimization approach combining constraint programming and local search using Berrera et al. (Berrera et al., 2003) deduced 20×20 energy matrix (we denote this model as BM). In a recent work, Shatabda et al. (2013a) presented a mixed heuristic local search algorithm for PSP and produced the state-of-the-art results using the BM model on 3D FCC lattice. Although the heuristics themselves are weaker than the BM energy model, their collective use in the random mixing fashion produce results better than the BM energy itself. In a previous work (Rashid et al., 2013d), we applied BM and HP energy models in a mixed manner within a GA framework and showed that hybridizing energies performs better than their individual performances.

In this work, we propose a graded as well as hybrid energy function with a genetic algorithm (GA) based sampling to develop an effective *ab initio* PSP tool. The graded energy-model strategically mixes 20×20 Miyazawa–Jernigan (MJ) contact-energy (Miyazawa and Jernigan, 1985; Berrera et al., 2003) with the simple 2×2 hydrophobic-polar (HP) contact-energy model (Lau and Dill, 1989), denoted as MH (MJ + HP $\rightarrow$ MH) in this paper. Specifically, we propose a hydrophobic-polar categorization of the HP model within a hydrophobic-core directed macro-mutation operator to explore the local benefits exhaustively while the GA sampling is guided by the MJ energy Matrix globally. While the fine grained details of the high resolution interaction energy matrix can become computationally prohibit, a low resolution energy model may effectively sample the search-space towards certain promising directions particularly emphasizing on the pair-wise contributions with large magnitudes—which we have implementation strategically via a macro mutation. Further, we use an enhanced genetic algorithm (GA) framework (Rashid et al., 2012a) for protein structure optimization on 3D face-centered-cube (FCC) lattice model. Prediction in the FCC lattice model can yield the densest protein core (Hoque et al., 2005) and the FCC lattice model can provide the maximum degree of freedom as well as the closest resemblance to the real or, high resolution folding within the lattice constraint. FCC orientation can therefore align a real protein into the closest conformation amongst the available lattice configurations (Hoque et al., 2007a).

On a set of standard benchmark proteins, our MH model guided GA, named as *MH_GeneticAlgorithm* (MH_GA), shows significant improvements in terms of interaction energies and root-mean-square deviations in comparison to the state-of-the-art search approaches (Ullah and Steinhöfel, 2010; Shatabda et al., 2013a; Torres et al., 2007) for the lattice based PSP models. For a fair comparison, we run Ullah and Steinhöfel (2010) and Shatabda et al. (2013a) using MJ energy model and in the result section, we compare our experimental results with the results produced by Ullah and Steinhöfel (2010) and Shatabda et al. (2013a). Further, we present an experimental analysis showing the effectiveness of using the hydrophobic polar categorization of the HP model to direct macro-mutation operation.

2. Background

Anfinsen's hypothesis (Anfinsen, 1973) and Levinthal's paradox (Levinthal, 1968) form the basis and the confidence of the *ab initio* approach, which inform that the protein structure prediction can be relied only on the amino acid sequence of the target protein as well as there should be a non-exhaustive pathway to obtain the native fold. Thus, we set our goal to model the folding process using on-lattice model. Further, it has been argued in Alm and Baker (1999) and Baker (2000), "... protein folding mechanisms and

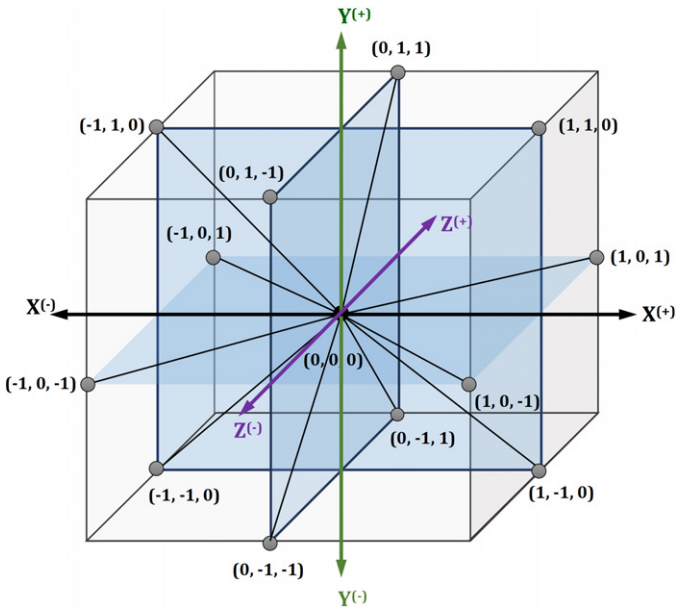

Fig. 1. A 3-dimensional face-centered-cubic lattice space. The 12 basis vectors of the neighbors of the origin (0, 0, 0) in a Cartesian coordinate system (Rashid et al., 2015).

Table 1
The 2 × 2 HP energy model.

	H	P
H	−1	0
P	0	0

landscapes are largely determined by the topology of the native state and are relatively insensitive to details of the interatomic interactions. This dependence on low resolution structural features, rather than on high resolution detail, suggests that it should be possible to describe the fundamental physics of the folding process using relatively low resolution models. . . The observation that protein folding mechanisms are determined primarily by low resolution topological features and not by high resolution details suggests that a simple theory incorporating features of the native state topology should be successful in predicting the broad outlines of folding reactions". A rigorous discussion on on-lattice models can be found in Istrail et al. (2009) and Bahi et al. (2013). Further, exploration of an astronomically large search space and the evaluation of the conformations using a real energy models are the big challenges often being computationally prohibitive, especially for sequences >200 residues long, whereas simplified models can aid in modeling and understanding the protein folding process feasibly. Next, we describe the model that we use.

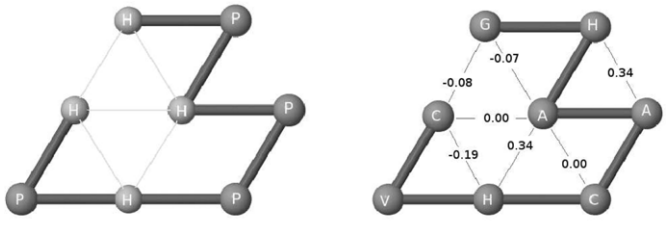

Fig. 2. On a lattice based model, (a) is showing H–H contact energy −4 (−1 ×4) using the HP model for a random sequence: *HPHPPHPH* (b) is showing the sum of the pairwise contact potentials 0.34 ((−0.08)+(−0.19)+(0.00) × 2+(−0.07)+(0.34) × 2) using the MJ model for a random sequence: *GHAACHVC*.

Table 2
The 20 × 20 MJ energy model by Miyazawa and Jernigan (1985). The bold-faced amino acids are Hydrophobic (H) and others are Polar (P).

	CYS	MET	PHE	ILE	LEU	VAL	TRP	TYR	ALA	GLY	THR	SER	GLN	ASN	GLU	ASP	HIS	ARG	LYS	PRO
CYS	−1.06																			
MET	0.19	0.04																		
PHE	−0.23	−0.42	−0.44																	
ILE	0.16	−0.28	−0.19	−0.22																
LEU	−0.08	−0.20	−0.30	−0.41	−0.27															
VAL	0.06	−0.14	−0.22	−0.25	−0.29	−0.29														
TRP	0.08	−0.67	−0.16	0.02	−0.09	−0.17	−0.12													
TYR	0.04	−0.13	0.00	0.11	0.24	0.02	−0.04	−0.06												
ALA	0.00	0.25	0.03	−0.22	−0.01	−0.10	−0.09	0.09	−0.13											
GLY	−0.08	0.19	0.38	0.25	0.23	0.16	0.18	0.14	−0.07	−0.38										
THR	0.19	0.19	0.31	0.14	0.20	0.25	0.22	0.13	−0.09	−0.26	0.03									
SER	−0.02	0.14	0.29	0.21	0.25	0.18	0.34	0.09	−0.06	−0.16	−0.08	0.20								
GLN	0.05	0.46	0.49	0.36	0.26	0.24	0.08	−0.20	0.08	−0.06	−0.14	−0.14	0.29							
ASN	0.13	0.08	0.18	0.53	0.30	0.50	0.06	−0.20	0.28	−0.14	−0.11	−0.14	−0.25	−0.53						
GLU	0.69	0.44	0.27	0.35	0.43	0.34	0.29	−0.10	0.26	0.25	0.00	−0.26	−0.17	−0.32	−0.03					
ASP	0.03	0.65	0.39	0.59	0.67	0.58	0.24	0.00	0.12	−0.22	−0.29	−0.31	−0.17	−0.30	−0.15	0.04				
HIS	−0.19	0.99	−0.16	0.49	0.16	0.19	−0.12	−0.34	0.34	0.20	−0.19	−0.05	−0.02	−0.24	−0.45	−0.39	−0.29			
ARG	0.24	0.31	0.41	0.42	0.35	0.30	−0.16	−0.25	0.43	−0.04	−0.35	0.17	−0.52	−0.14	−0.74	−0.72	−0.12	0.11		
LYS	0.71	0.00	0.44	0.36	0.19	0.44	0.22	−0.21	0.14	0.11	−0.09	−0.13	−0.38	−0.33	−0.97	−0.76	0.22	0.75	0.25	
PRO	0.00	−0.34	0.20	0.25	0.42	0.09	−0.28	−0.33	0.10	−0.11	−0.07	0.01	−0.42	−0.18	−0.10	0.04	−0.21	−0.38	0.11	0.26

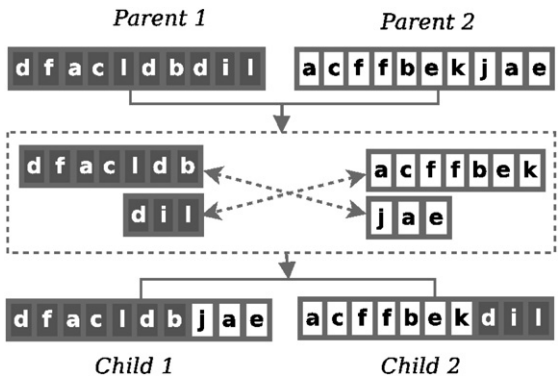

Fig. 3. Typical crossover operator: exchanging parts and forming new chromosomes.

2.1. Simplified model

In our simplified model, we use 3D FCC lattice points to map the amino acids of a protein sequence. In the mapping, each amino acid of the sequence, occupies a point on the lattice to form a continuous chain of a self-avoiding-walk. We apply the MJ energy matrix in conjunction with the HP energy model in a genetic algorithm framework for PSP. The FCC lattice, the HP and MJ energy models, and the GA are briefly described below.

2.1.1. FCC lattice

The FCC lattice has the highest packing density compared to the other existing lattices (Hales, 2005). Thus, FCC model can provide maximum degree of freedom within a constrained space. In FCC, each lattice point (the origin in Fig. 1) has 12 neighbors with closest possible distance having 12 *basis vectors* as follows:

$$v_1 = (1,1,0) \quad v_4 = (-1,-1,0) \quad v_7 = (-1,1,0) \quad v_{10} = (0,1,-1)$$
$$v_2 = (1,0,1) \quad v_5 = (-1,0,-1) \quad v_8 = (1,-1,0) \quad v_{11} = (1,0,-1)$$
$$v_3 = (0,1,1) \quad v_6 = (0,-1,-1) \quad v_9 = (-1,0,1) \quad v_{12} = (0,-1,1)$$

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.

2.1.2. HP energy model

Based on the hydrophobic property, the 20 amino acids which are the constituents of all proteins, are broadly divided into two categories: (a) hydrophobic amino acids (*Gly*, *Ala*, *Pro*, *Val*, *Leu*, *Ile*, *Met*, *Phe*, *Tyr*, *Trp*) are denoted as H; and (b) hydrophilic or polar amino acids (*Ser*, *Thr*, *Cys*, *Asn*, *Gln*, *Lys*, *His*, *Arg*, *Asp*, *Glu*) are denoted as P. In the 2×2 HP model (Lau and Dill, 1989), when two non-consecutive hydrophobic amino acids become topologically neighbors, they contribute a certain amount of negative energy, which for simplicity is considered as -1 (Table 1). The total energy E_{HP} (Eq. (1)) of a conformation based on the HP model becomes the sum of the contributions over all pairs of the non-consecutive hydrophobic amino acids (Fig. 2a).

$$E_{HP} = \sum_{i<j-1} c_{ij} \times e_{ij} \tag{1}$$

where $c_{ij} = 1$ if amino acids at positions i and j in the sequence are non-consecutive but topological neighbors on the lattice, otherwise $c_{ij} = 0$. The $e_{ij} = -1$ if the ith and jth amino acids are both hydrophobic, otherwise $e_{ij} = 0$.

Fig. 4. Typical mutation operator: mutating one point into some other point.

2.1.3. MJ energy model

By analyzing crystallized protein structures, Miyazawa and Jernigan (1985) statistically deduced a 20×20 energy matrix (better known as MJ energy model) that considers residue contact propensities between the amino acids. BM is a similar energy matrix as MJ deduced by Berrera et al. (Berrera et al. (2003)) by calculating empirical contact energies on the basis of information available from a set of selected protein structures and following the quasi-chemical approximation. In this work, we use MJ energy model. The total energy E_{MJ} (Eq. (2)) of a conformation based on the MJ energy model is the sum of the contributions of all of the non-consecutive amino acid pairs that are topological neighbors (Fig. 2b).

$$E_{MJ} = \sum_{i<j-1} c_{ij} \times e_{ij} \tag{2}$$

where $c_{ij} = 1$ if amino acids at positions i and j in the sequence are non-consecutive neighbors on the lattice, otherwise $c_{ij} = 0$; and e_{ij} is the empirical energy value between the ith and jth amino acid pair specified in the MJ energy matrix as shown in Table 2.

Algorithm 1: The standard genetic algorithm

```
/* INPUT: Protein sequence, Crossover rate, Mutation rate, Population
   size                                                              */
/* OUTPUT: Global best solution                                      */
1 initialize population with encoded protein sequences as chromosomes (individuals);
2 evaluate population;
3 repeat
4 |   Select the best-fit individuals for reproduction;
5 |   Breed new individuals using crossover and mutation operations according to the rates;
6 |   Evaluate the new individuals;
7 |   Replace least-fit individuals with new best-fit individuals;
8 until (termination criteria);
```

2.2. Genetic algorithms

GAs (Holland, 1975) are a family of population-based search algorithms which can be applied for PSP as an optimization problem. The outline of GA as given in Algorithm 1, follows simple steps: Line 1 initializes the population; the Line 2 evaluates the solutions to rank them by relative quality; and the Lines 4–7 are repeating on generating, evaluating and replacing the least-fitted off-springs

within the population until the termination criteria arises. For the coding scheme, non-isomorphic encoding (Hoque et al., 2006) has been applied and the $v_1, \ldots, v_{12}$ (in Fig. 1) can be thought of be renamed as $a, \ldots, l$ respectively.

A typical crossover operator randomly splits two solutions at a randomly selected crossover point and exchanges the parts between them (Fig. 3) and a typical mutation operator alters a solution at a random point (Fig. 4). In the case of PSP, conformations are regarded as solutions of a GA population.

3. Methods

This section describes the proposed MH_GA framework along with the implementation level detail. We implemented the framework in Java (J2EE), using Rocks clusters. The code for MH based GA is freely available online.[1]

3.1. The primitive operators implemented in the GA framework

The primitive operators that we implemented within the MH_GA framework are crossover (Fig. 5a), rotation mutation (Fig. 5b), diagonal move (Fig. 5c), pull moves (Fig. 5d), and tilt moves (Fig. 5e). The Rotation, diagonal move, pull moves and tilt moves are implemented as mutation operators.

1. *Crossover:* At a given crossover point (dotted circle in (Fig. 5a), two parent conformations exchange their parts and generate two children. The success rate of crossover operator decreases with the increase of the compactness of the structure.
2. *Rotation:* One part of a given conformation is rotated around a selected point (Fig. 5b). This move is mostly effective at the beginning of the search.
3. *Diagonal move:* Given three consecutive amino acids at lattice points *A*, *B*, and *C*, a diagonal move at position *B* takes the corresponding amino acid diagonally to a free position (Fig. 5c). The diagonal moves are very effective on FCC lattice (Cebrián et al., 2008; Dotú et al., 2011) points.
4. *Pull moves:* The amino acids at points *A* and *B* are pulled to the free points (Fig. 5d) and the connected amino acids are pulled as well to get a valid conformation. The pull moves (Lesh et al., 2003) are local, complete, and reversible. These are very effective especially when the conformation is compact.
5. *Tilt moves:* Two or more consecutive amino acids connected in a straight line are moved by a tilt move to immediately parallel lattice positions (Hoque, 2008). The tilt-moves pull the conformation from both sides until a valid conformation is found. In Fig. 5e, the amino acids at points *C* and *D* are moved and subsequently other amino acids from both sides are moved as well.

3.2. Genetic algorithm framework

The *pseudocode* of MH_GA framework is presented in Algorithm 2. It uses a set of primitive operators Fig. 5) in an exhaustive generation approach to diversify the search, a hydrophobic core-directed macro-mutation operator to intensify the search, and a random-walk algorithm to recover from the stagnation. Like other search algorithms, it requires initializing the population and the solutions need to be evaluated in each iteration.

Algorithm 2: MH_GeneticAlgorithm (MH_GA)

```
/* INPUT: Protein/Amino acid sequence, popSize : Population size; opR:
   Operator selection probabilities                                    */
/* OUTPUT: Global best conformation                                    */
/* VARIABLE: op: Operators; c, c': Conformations; curP, newP: Current
   and new populations; mmCount: Macro-mutation counter; rwCount:
   Non-improving random-walk counter                                   */
1  curP ←initialise ;
2  repeat
3  |  op ←, selectOperator (opR);
4  |  if (op is crossover) then
   |  |  /* ** go for crossover                                        */
5  |  |  while (¬ full (newP)) do
6  |  |  |  c, c' ← randomConfs (curP);
7  |  |  |  newP.add (doCrossover);
8  |  |  end
9  |  else if (op is mutation) then
   |  |  /* ** go for mutation                                        */
10 |  |  foreach (c ∈ curP) do
11 |  |  |  newP.add (doMutation);
12 |  |  end
13 |  else
   |  |  /* ** go for macro-mutation                                  */
14 |  |  foreach (c ∈ curP) do
15 |  |  |  newP.add (doHCDMacroMutation);
16 |  |  end
17 |  end
18 |  if (¬ improved (newP,rwCount)) then
19 |  |  newP ←goRandomWalk ;
20 |  end
21 |  curP ← newP;
22 until (termination criteria);
23 return bestConformation (curP);
```

[1] Download the JAR file from: http://cs.uno.edu/ tamjid/Software/MH_GA/JarFiles.zip.

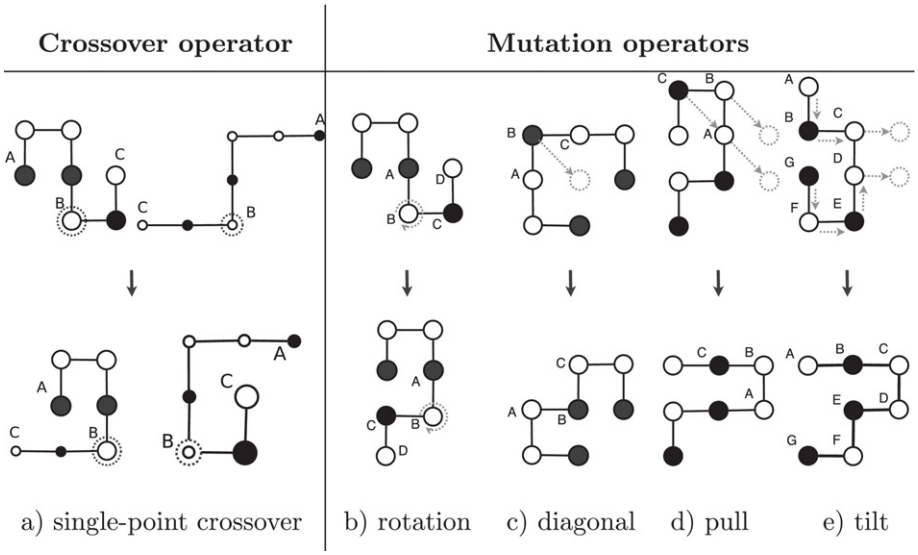

Fig. 5. The primitive operators used in our genetic algorithms. The crossover operator applied on two parent conformations to exchange their parts to generate two child conformations (as shown in a) and the mutation operators are applied on single conformation to generate single child conformation (as shown in b–e). The operators are implemented in 3D space, however, for simplification and easy understanding the figures are presented in 2D space. The black solid circles represent the hydrophobic amino acids and others are polar.

The algorithm initializes (Algorithm 2: Line 7) the current population with randomly generated individuals. At each generation, it selects a genetic operator based on a given probability distribution to use through the generation (Algorithm 2: Line 9). In fact, we select the operators randomly by giving equal opportunities to all operators. The selected operator is used in an exhaustive manner (Algorithm 2: Lines 11–12 or Lines 14–16) to obtain all conformations in the new population. We ensure that no duplicate conformation is added to the new population. The *add*() method (Line 12 or 16 in Algorithm 2) takes care of adding the non-duplicate conformations to the new population. For a given number of generations, if the best conformation in the new population is not better than the best in the current population, our algorithm triggers a random-walk technique (Algorithm 2: Line 18) to diversify the new population. Nevertheless, after each generation, the new population becomes the current population (Algorithm 2: Line 19); and the search continues. Finally, the best conformation found so far is returned (Algorithm 2: Line 20). Along with MJ potential matrix, the HP energy model is used during move selection by the macro-mutation operator. The macro-mutation operator is used as other mutation operators Fig. 5b–e) in MH_GA. The details of initialization, evaluation of fitness, exhaustive generation, macro-mutation and stagnation recovery schemes are presented below.

3.2.1. Initialization

Our algorithm starts with a feasible set of conformation known as population. We generate initial conformations following a self-avoiding walk on FCC lattice points. The *pseudocode* of the algorithm is presented in Algorithm 3. 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 neighboring free lattice point. The mapping proceeds until a self-avoiding walk is found for the whole protein sequence.

Algorithm 3: initialise

```
/* Is called from Algorithm 2 in Line 1                                   */
/* INPUT: Protein/Amino acid sequence, FCC basis vectors, popSize :
   Population size                                                        */
/* OUTPUT: Initial population                                             */
/* VARIABLE: AA : Array of amino acid; c : Conformations; point :
   Unoccupied point on 3D FCC Lattice space                               */
1  for (p = 1; p ≤ popSize; p++) do
2      AA[0] ⟵ aminoAcid (0,0,0);
3      for a number of times do
4          for (i =1 to seqLength-1) do
5              j ⟵ getRandom (12);
6              point ⟵ AA[i − 1] + basisVector[j];
7              if point is not free then
8                  break;
9              else
10                 AA[i] ⟵ aminoAcid (point);
11             end
12         end
13     end
14     if full structure found then
15         c.AminoAcid⟵AA [ ];
16     else
17         c ⟵ a deterministic structure;
18     end
19     c.fitness ⟵evaluate (c.aminoAcid);
20     initPop.add (c);
21 end
22 return initPop
```

3.2.2. Evaluate the fitness

For each iteration, the conformation is evaluated by calculating the contacts (topological neighbor) potentials where the two amino acids are non-consecutive. The pseudo-code in Algorithm 4 presents the algorithm for calculating the interaction energy of a given conformation. The contact potentials are found in MJ potential matrix (Miyazawa and Jernigan, 1985) (see Table 2).

Algorithm 4: evaluate

```
/* Is called from Algorithm 3 in Line 19                                  */
/* INPUT: MJ energy matrix(20 × 20), AA : Array of amino acid             */
/* OUTPUT: Fitness of the structure                                       */
/* VARIABLE: seqLength : Sequence length; pointI, pointj : Occupied point
   on 3D FCC Lattice space                                                */
1  fitness ⟵ 0
2  for (i =0 to seqLength − 1) do
3      for (j =i +2 to seqLength − 1) do
4          pointI ⟵ AA [i];
5          pointJ ⟵ AA [j];
6          sqrD ⟵ getSqrDist (pointI, pointJ);
7          if sqrD =2 then
8              fitness ⟵ fitness+E_bm[i][j];
9          end
10     end
11 end
12 return fitness;
```

3.2.3. Exhaustive generation

Unlike standard genetic algorithm, in MH_GA, the randomness is reduced significantly by applying exhaustive generation approach. For mutation operators, MH_GA adds one resultant conformation to the new population that corresponds to *each* conformation in the current population. Operators are applied to all possible point

(Algorithm 5) 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 (Algorithm 6) 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.

3.2.4. Macro-mutation operator

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 (Yue and Dill, 1993). H-core formation is an important objective of HP based PSP. Macro-mutation operator is a composite operator (Fig. 6) that uses a series of diagonal-moves (Fig. 5c) on a given conformation to build the H-core around the hydrophobic-core-center (HCC). The macro-mutation squeezes the conformation and quickly forms the H-core. In MH_GA, macro-mutation is used as other mutation operators. Algorithm 7 presents the *pseudocode* of macro-mutation algorithm.

Algorithm 5: doMutation

```
/* Is called from Algorithm 2 in Line 11                                  */
/* INPUT: conf : Conformation                                             */
/* OUTPUT: Best mutated conformation                                      */
/* VARIABLE: c : Conformation; offspring : List of type conformation      */
1 offspring.add(conf);
2 foreach (1 ≤ pos ≤ seqLength) do
3 |   c ← applyOperator(conf, pos);
4 |   offspring.add(c);
5 end
6 return bestConformation(offspring);
```

Algorithm 6: doCrossover

```
/* Is called from Algorithm 2 in Line 7                                   */
/* INPUT: c1 and c2 : Conformations                                       */
/* OUTPUT: Best two conformations after crossover                         */
/* VARIABLE: c, c′ : Conformations; offspring : List of type conformation */
1 offspring.add(c1, c2);
2 foreach (1 ≤ pos ≤ seqLength) do
3 |   c, c′ ← applyOperator(c1, c2, pos);
4 |   offspring.add(c, c′);
5 end
6 return best2Conformations(offspring);
```

Algorithm 7: doHCDMacroMutation

```
/* Is called from Algorithm 2 in Line 15                                  */
/* INPUT: HP energy matrix(2 × 2), C: Conformation; repeat: Loop counter  */
/* OUTPUT: Mutated conformation                                           */
/* VARIABLE: T: Either hydrophobic (H) or polar (P); AA: Array of amino   */
   acid                                                                   */
1  AA [ ]⟵C.AminoAcid[ ];
2  for i = 1 to repeat do
3  |   T ⟵ P if bernoulli(p), else H
4  |   AA[j] : jth amino acid in conformation
5  |   point: unoccupied new position for AA[j]
6  |   hcc ⟵findHCC ()
7  |   foreach j : typeOf(AA[j]) = T do
8  |   |   d_old ⟵getDistance (AA[j],hcc)
9  |   |   if T = P then
10 |   |   |   point ⟵findFreePoint (AA[j])
11 |   |   |   applyDiagonalMove(AA[j], point)
12 |   |   end
13 |   |   else
14 |   |   |   point ⟵findFreePoint (AA[j])
15 |   |   |   d_new ⟵getDistance (point,hcc)
16 |   |   |   if d_new ≤d_old then
17 |   |   |   |   applyDiagonalMove(AA[j], point)
18 |   |   |   |   break
19 |   |   |   end
20 |   |   end
21 |   end
22 end
23 C.AminoAcid[ ]⟵AA [ ];
24 return C
```

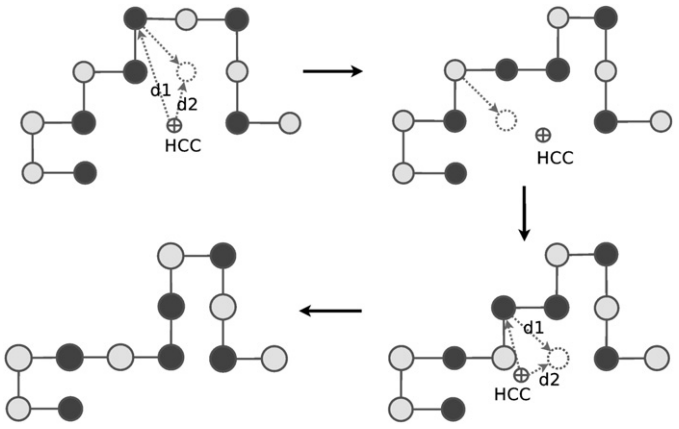

Fig. 6. A macro-mutation operator repeatedly used diagonal moves. The moves of an amino acid are guided by the distance of current position (d_1) and the distance of target position (d_2) from the HCC. The operator is implemented in 3D space, however, for simplification and easy understanding, the figures are drawn in 2D space.

In macro-mutation, the HCC is calculated by finding arithmetic means of *x*, *y*, and *z* coordinates of all H amino acids. In macro-mutation, for a given number of iterations, diagonal moves apply repeatedly 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 *Bernoulli* distribution (Algorithm 7: Line 2) with probability *p* (intuitively we use $p = 20\%$ for P-type amino acids). For a P-type amino acid, the first successful diagonal move is considered. However, 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.

3.3. Stagnation recovery

Like other search algorithm, GA can get stuck in the local minima or, can be stalled. Stall condition can occur when similarities with the chromosomes in GA increases heavily and the operators are unable to produce better diverse solutions. Further, with the PSP search, resulting solutions become phenotypically compact which reduce the likelihood of producing better solution from the population due to harder self-avoid-walk (SAW) constraints (Higgs et al., 2012b; Hoque et al., 2007b, 2011). It would rather require very intelligent moves to reform into another competitive compact SAW. To deal with such situation, we apply the following two actions:

3.3.1. Removing duplicates

In genetic algorithm it has been observed that with increasing generations, the similarity among the individuals within the population increases. In worst case scenario, all the individuals become similar and forces the search to stall in the local minima. In our approach, we remove duplicates from each generation to maintain the diversity of the population. During exhaustive generation, we check the existence of the newly generated child in the new population. If it does not exist then the new solution is added to the new population list. Our approach reduces the frequency of stagnations.

3.3.2. Applying random-walk

Sometimes, early convergence leads the search towards the stagnation situation. In the HP energy model, premature H-cores are observed at local minima. To break these H-cores, in MH_GA (Algorithm 2: Line 18), a random-walk algorithm (Algorithm 8) is applied. This algorithm uses pull moves (Lesh et al., 2003) (as shown in Fig. 5d) to break the H-core. We use pull-moves because they are complete, local, and reversible. Successful pull moves never generate infeasible conformations. During pulling, energy level and structural diversification are observed to maintain balance among these two. We allow energy level to change within 5–10% that changes the structure from 10% to 75% of the original. We try to accept the conformation that is close to the current conformation in terms of the energy level but as far as possible in structural diversity, and which is determined by the function *checkDiversity*() in Algorithm 8 at Line 5. For genetic algorithm, random-walk is very effective (Rashid et al., 2012b) to recover from stagnation.

Algorithm 8: goRandomWalk

```
/* Is called from Algorithm 2 in Line 19                                  */
/* INPUT: inPop:  Current population; pct:  Changed percentage (%)        */
/* OUTPUT:  New diverged population                                       */
/* VARIABLE:  outPop:  New diverged population;        AA: Array of
   amino acid; c, c' : Conformations                                      */
1  foreach (c ∈ inPop) do
2      isFound ⟵ false;
3      AA [ ]⟵c.AminoAcid[ ];
4      while (¬isFound) do
5          for (i = 1; i ≤ seqLength & ¬isFound; i++) do
6              applyPullMove(AA[i]);
7              c'.AminoAcid[ ]⟵AA [ ];
8              isFound ⟵ checkDiversity (c,c',pct);
9          end
10     end
11     outPop.add (c');
12 end
13 return outPop
```

The complete flow of MH_GeneticAlgorithm (Algorithm: 2) is graphically presented in Fig. 7. Further, it describes the steps taken within macro mutation procedure (Algorithm: 7).

4. Performance evaluation

To compare and evaluate the performance of the proposed PSP predictor with respect to the state-of-the-art approaches, we used the measures *Relative Improvement (RI)* and *RMSD* comparisons. They are defined below:

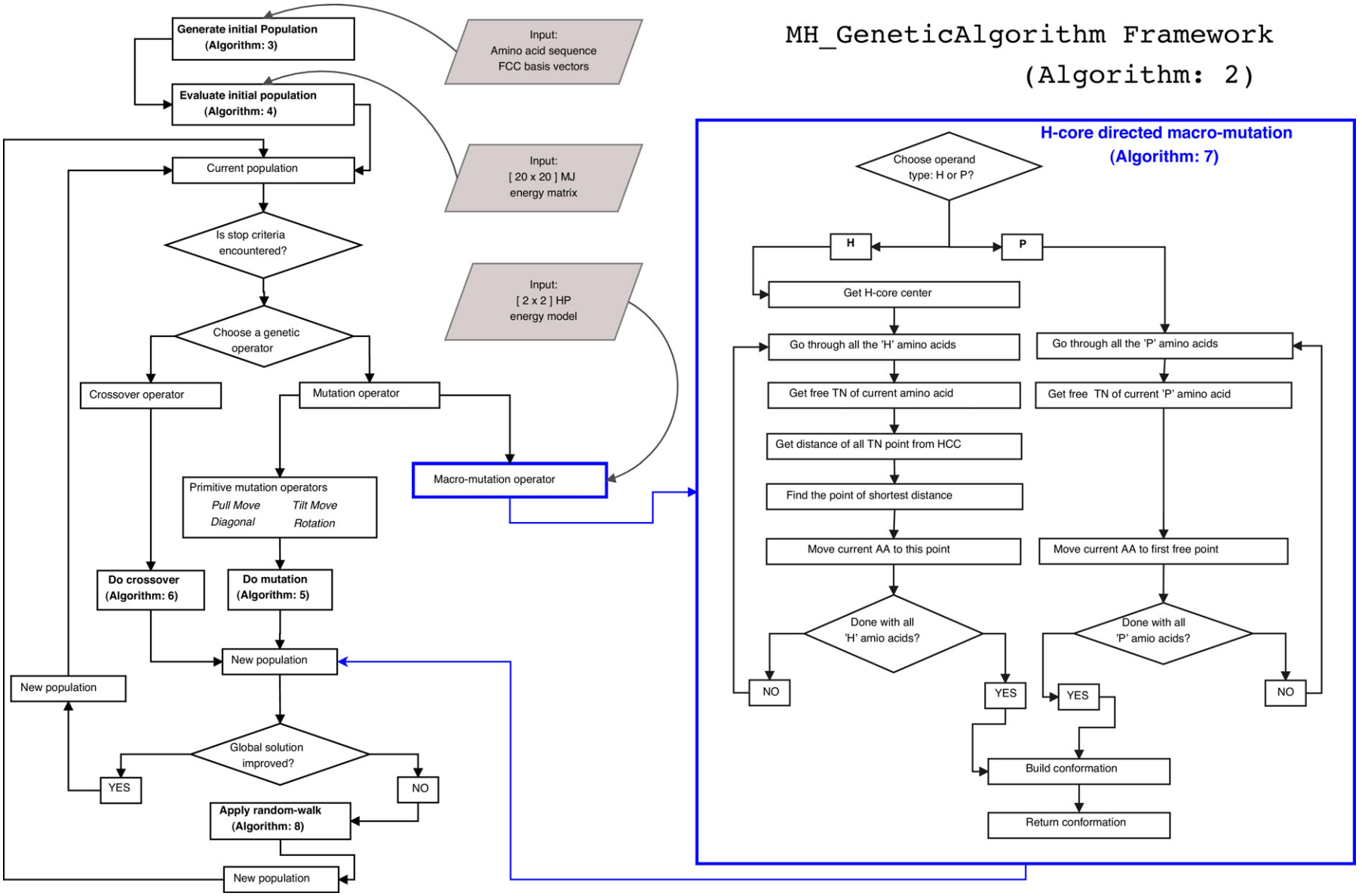

Fig. 7. A complete overview of our algorithmic approach. The macro-mutation procedure is described step by step (inside the blue box). The procedural sub blocks are marked in bold along with the corresponding labels of the algorithms described above. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

4.1. Relative Improvement (RI)

The difficulty to improve energy level is increased as the predicted energy level approaches to a known lower bound of a given protein. 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). The RI computes the relative improvements that our algorithm (target,t) achieved w.r.t. the state-of-the-art approaches (reference, r).

For each protein, the relative improvement of the target (t) w.r.t. the reference (r) is calculated using the formula in Eq. (3), where E_t and E_r denote the average energy values achieved by target and reference respectively.

$$\text{RI} = \frac{E_t - E_r}{E_r} \times 100\% \tag{3}$$

4.2. RMSD comparison

The root mean square deviation (RMSD) is frequently used to measure the differences between values predicted by a model and the values actually observed. We compare the predicted structures obtained by our approach with the state-of-the-art approaches by measuring the root-mean-square w.r.t. the native structures from PDB. For any given structure the root-mean-square is calculated using Eq. (4),

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(d_{ij}^{\text{p}} - d_{ij}^{\text{n}})^2}{n \times (n-1)/2}} \tag{4}$$

where d_{ij}^{p} and d_{ij}^{n} denote the distances between ith and jth amino acids respectively in the predicted structure and the native structure of the protein. The average distance between two α-carbons in native structure is 3.8Å. To calculate root-mean-square, the distance between two neighbor lattice points is considered as 3.8Å.

5. Results and discussion

In this section, we discuss the obtained results along with the comparison of the performance of MH_GeneticAlgorithm with the other state-of-the-art results (Torres et al., 2007; Ullah and Steinhöfel, 2010; Shatabda et al., 2013a). Further, we present an analysis of the results.

5.1. Benchmark

In our experiment, the protein instances are taken from the literatures. The first seven proteins (*4RXN*, *1ENH*, *4PTI*, *2IGD*, *1YPA*, *1R69*, and *1CTF*) in Table 3 are taken from Ullah and Steinhöfel (2010) and Shatabda et al. (2013a), and the next five proteins (*3MX7*, *3NBM*, *CMQO*, *3MRO*, and *3PNX*) are taken from Shatabda et al. (2013a). The two other protein instances in Table 5 (*2J61* and *2HFQ*) are taken from Torres et al. (2007).

5.2. Comparing with the state-of-the-art

In the literature we found very few works (Kapsokalivas et al., 2009; Torres et al., 2007) that used 20 × 20 MJ potential-matrix (Miyazawa and Jernigan, 1985) for protein structure prediction on 3D FCC lattice. However, Torres et al. (2007) used 3D HCP lattice and Kapsokalivas et al. (2009) used 3D cubic lattice in their works for protein mapping. In other works, Ullah and Steinhöfel (2010) and Shatabda et al. (2013a) used 3D FCC lattice with 20 × 20 empirical energy matrix by Berrera et al. (2003). In fact, we do not have any state-of-the-art results available for similar model to compare free energy level in a straight way. Therefore, we ran the algorithms used in Ullah and Steinhöfel (2010) and Shatabda et al. (2013a) using the MJ energy model (Miyazawa and Jernigan, 1985) to compare our results. However, the constraint programming based hybrid approach (Ullah and Steinhöfel, 2010) failed to get any solution for most of the large-sized proteins. In such cases, in Table 4, the results are denoted by *n/a*.

In Table 4, we present interaction energy values in two different formats: the global lowest interaction energy (Column *Best*) and the average (Column *Avg*) of the lowest interaction energies obtained from 50 different runs. In case of the global best energy, our approach outperforms the state-of-the-art approaches in Ullah and Steinhöfel (2010) and Shatabda et al. (2013a) on 9 out of 12 benchmark proteins. However, in case of average energy, our approach outperforms both of the approaches on 10 out of 12 benchmark proteins. Based on the experimental results, the performance hierarchy of the approaches used to validate our MH_GA is shown in Fig. 8.

5.2.1. Outcome based on Relative Improvement (RI)

From the Column RI of Table 4, we see that for 2 proteins our GA fail to improve over the state-of-the-art. However, for other 10 proteins it improves the average interaction energy level ranging from 0.10% to 26.58% for different proteins.

Further, in Table 5, we present another two benchmark proteins taken from a GA based approach (Torres et al., 2007). From the authors of Torres et al. (2007), we tried to get their implemented codes so that we can run that by ourselves. However, we failed to receive any response from the authors. Therefore, we present the reported values. For fair comparison, we compare the results by generation-wise instead of by running-time.

5.2.2. Outcome based on RMSD comparison

We calculate RMSD of a structure that corresponds to the lowest MJ interaction energy for a particular run. The reported RMSD values in Table 6 are the global minimum of 50 runs. In Tables 5 and 6, the bold-faced RMSD values indicate the winners for the corresponding proteins.

In Table 7, we present corresponding MJ energy values for global minimum RMSD and corresponding RMSD values for global minimum MJ energy values over 50 runs for each proteins on identical settings. The experimental results show that the global minimum energy in our experiment does not produce minimum RMSD value.

5.3. Result analysis

The MJ energy model actually implicitly bear the characteristic of hydrophobicity. The matrix values present some variations within amino acids of the same class (H or P). A partition algorithm such as 2-means clustering algorithm easily reveals the H–P partitioning within the MJ model. Given this knowledge, we study the effect of explicitly using hydrophobic property within our GA.

5.3.1. Effect of HP in MH model

Our macro-mutation operator biases the search towards a hydrophobic core by applying a series of diagonal moves and thus achieves improvements in terms of MJ energy values of the output conformations. We implemented three different versions of our genetic algorithm.

1. *MH:* This version is our final algorithm that we described in detail, and used in presenting our main results in Table 4 and in comparing with the state-of-the-art results. To reiterate, this version uses the MJ energy model for search and energy reporting, and hydrophobicity knowledge in the macro-mutation

Table 3
The benchmark proteins used in our experiments.

ID	Len	Protein sequence
4RXN	54	MKKYTCTVCGYIYNPEDGDPDNGVNPGTDFKDIPDDWVCPLCGVGKDQFEEVEE
1ENH	54	RPRTAFSSEQLARLKREFNENRYLTERRRQQLSSELGLNEAQIKIWFQNKRAKI
4PTI	58	RPDFCLEPPYTGPCKARIIRYFYNAKAGLCQTFVYGGCRAKRNNFKSAEDCMRTCGGA
2IGD	61	MTPAVTTYKLVINGKTLKGETTTKAVDAETAEKAFKQYANDNGVDGVWTYDDATKTFTVTE
1YPA	64	MKTEWPELVGKAVAAAKKVILQDKPEAQIIVLPVGTIVTMEYRIDRVRLFVDKLDNIAQVPRVG
1R69	69	SISSRVKSKRIQLGLNQAELAQKVGTTQQSIEQLENGKTKRPRFLPELASALGVSVDWLLNGTSDSNVR
1CTF	74	AAEEKTEFDVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAPAALKEGVSKDDAEALKKALEEAGAEVEVK
3MX7	90	MTDLVAVWDVALSDGVHKIEFEHGTTSGKRVVYVDGKEEIRKEWMFKLVGKETFYVGAAKTKATINIDAISGFA YEYTLEINGKSLKKYM
3NBM	108	SNASKELKVLVLCAGSGTSAQLANAINEGANLTEVRVIANSGAYGAHYDIMGVYDLIILAPQVRSYYREMKVDA ERLGIQIVATRGMEYIHLTKSPSKALQFVLEHYQ
3MQO	120	PAIDYKTAFHLAPIGLVLSRDRVIEDCNDELAAIFRCARADLIGRSFEVLYPSSDEFERIGERISPVMIAHGSY ADDRIMKRAGGELFWCHVTGRALDRTAPLAAGVWTFEDLSATRRVA
3MRO	142	SNALSASEERFQLAVSGASAGLWDWNPKTGAMYLSPHFKKIMGYEDHELPDEITGHRESIHPDDRARVLAALKA HLEHRDTYDVEYRVRTRSGDFRWIQSRGQALWNSAGEPYRMVGWIMDVTDRKRDEDALRVSREELRRL
3PNX	160	GMENKKMNLLLFSGDYDKALASLIIANAAREMEIEVTIFCAFWGLLLLRDPEKASQEDKSLYEQAFSSLTPREA EELPLSKMNLGGIGKKMLLEMMKEEKAPKLSDLLSGARKKEVKFYACQLSVEIMGFKKEELFPEVQIMDVKEYL KNALESDLQLFI
2J6A	135	MKFLTTNFLKCSVKACDTSNDNFPLQYDGSKCQLVQDESIEFNPEFLLNIVDRVDWPAVLTVAAELGNNALPPT KPSFPSSIQELTDDDMAILNDLHTLLLQTSIAEGEMKCRNCGHIYYIKNGIPNLLLPPHLV
2HFQ	85	MQIHVYDTYVKAKDGHVMHFDVFTDVRDDKKAIEFAKQWLSSIGEEGATVTSEECRFCHSQKAPDEVIEAIKQN GYFIYKMEGCN
3MSE	180	GISPNVLNNMKSYMKHSNIRNIIINIMAHELSVINNHIKYINELFYKLDTNHNGSLSHREIYTVLASVGIKKWD INRILQALDINDRGNITYTEFMAGCYRWKNIESTFLKAAFNKIDKDEDGYISKSDIVSLVHDKVLDNNDIDNFF LSVHSIKKGIPREHIINKISFQEFKDYMLSTF
3MR7	189	SNAERRLCAILAADMAGYSRLMERNETDVLNRQKLYRRELIDPAIAQAGGQIVKTTGDGMLARFDTAQAALRCA LEIQQAMQQREEDTPRKERIQYRIGININGDIVLEDGDIFGDAVNVAARLEAISEPGAICVSDIVHQITQDRVSE PFTDLGLQKVKNITRPIRVWQWVPDADRDQSHDPQPSHVQH
3MQZ	215	SNAMSVQTIERLQDYLLPEWVSIFDIADFSGRMLRIRGDIRPALLRLASRLAELLNESPGPRPWYPHVASHMRRR VNPPPETWLALGPEKRGYKSYAHSGVFIGGRGLSVRFILKDEAIEERKNLGRWMSRSGPAFEQWKKKVGDLRDFG PVHDDPMADPPKVEWDPRVFGERLGSLKSASLDIGFRVTFDTSLAGIVKTIRTFDLLYAEAEKGS
3NO3	238	GKDNTKVIAHRGYWKTEGSAQNSIRSLERASEIGAYGSEFDVHLTADNVLVVYHDNDIQGKHIQSCTYDELKDLQ LSNGEKLPTLEQYLKRAKKLKNIRLIFELKSHDTPERNRDAARLSVQMVKRMKLAKRTDYISFNMDACKEFIRLC PKSEVSYLNGELSPMELKELGFTGLDYHYKVLQSHPDWVKDCKVLGMTSNVWTVDDPKLMEEMIDMGVDFITTDL PEETQKILHSRAQ
3NO7	248	MGSDKIHHHHHHENLYFQGMTFSKELREASRPIIDDIYNDGFIQDLLAGKLSNQAVRQYLRADASYLKEFTNIYA MLIPKMSSMEDVKFLVEQIEFMLEGEVEAHEVLADFINEPYEEIVKEKVWPPSGDHYIKHMYFNAFARENAAFTI AAMAPCPYVYAVIGKRAMEDPKLNKESVTSKWFQFYSTEMDELVDVFDQLMDRLTKHCSETEKKEIKENFLQSTI HERHFFNMAYINEKWEYGGNNNE
3ON7	280	GMKLETIDYRAADSAKRFVESLRETGFGVLSNHPIDKELVERIYTEWQAFFNSEAKNEFMFNRETHDGFFPASIS ETAKGHTVKDIKEYYHVYPWGRIPDSLRANILAYYEKANTLASELLEWIETYSPDEIKAKFSIPLPEMIANSHKT LLRILHYPPMTGDEEMGAIRAAAHEDINLITVLPTANEPGLQVKAKDGSWLDVPSDFGNIIINIGDMLQEASDGY FPSTSHRVINPEGTDKTKSRISLPLFLHPHPSVVLSERYTADSYLMERLRELGVL

Table 4
The energy values are obtained from different algorithms for the specified energy models. The average values are calculated over 50 different runs. The bold-faced values indicate the winner (the lower the better).

Protein details			The state-of-the-art						Our approach			
			Hybrid (Ullah and Steinhöfel, 2010)			Local Search (Shatabda et al., 2013a)			The MH_GA			
			MJ energy		Time	MJ energy		Time	MJ energy		Time	RI
Seq	*Size*	*H*	*Best*	*Avg*	*Avg*	*Best*	*Avg*(*r*)	*Avg*	*Best*	*Avg*(*t*)	*Avg*	Over (Shatabda et al., 2013a)
4RXN	54	27	−32.61	−30.94	1:02:12	−33.33	−31.21		**−36.36**	**−33.60**		7.66%
1ENH	54	19	−35.81	−35.07	1:02:03	−29.03	−28.18		**−38.39**	**−35.67**		26.58%
4PTI	58	32	−32.07	−29.37	1:01:26	−31.16	−28.33		**−35.65**	**−31.01**		9.46%
2IGD	61	25	**−38.64**	−32.54	1:43:08	−32.36	−28.29	1:00:00	−36.49	**−33.75**	1:00:00	19.30%
1YPA	64	38	*n/a*	*n/a*		−33.33	−32.15		**−40.14**	**−36.33**		13.00%
1R69	69	30	−34.2	−31.85	1:07:32	−33.35	−32.20		**−40.85**	**−36.28**		12.67%
1CTF	74	42	−38	−35.28	1:37:44	−45.83	−40.94		**−51.5**	**−47.29**		15.51%
3MX7	90	44	*n/a*	*n/a*		−44.81	−42.32		**−56.32**	**−50.95**		20.39%
3NBM	108	56	*n/a*	*n/a*		−52.44	−49.51		**−53.66**	**−49.9**		0.79%
3MQO	120	68	*n/a*	*n/a*		**−64.04**	**−58.84**	1:00:00	−62.25	−54.56	1:00:00	*no RI*
3MRO	142	63	*n/a*	*n/a*		−87.38	−82.24		**−90.05**	**−82.32**		0.10%
3PNX	160	84	*n/a*	*n/a*		**−103.04**	**−96.86**		−102.55	−88.06		*no RI*
3MSE	180	83	*n/a*	*n/a*		*n/a*	*n/a*		−92.61	−84.60		*n/a*
3MR7	189	88	*n/a*	*n/a*		*n/a*	*n/a*		−93.65	−83.93		*n/a*
3MQZ	215	115	*n/a*	*n/a*		*n/a*	*n/a*		−104.29	−95.22	2:00:00	*n/a*
3NO3	238	102	*n/a*	*n/a*		*n/a*	*n/a*		−122.97	−108.70		*n/a*
3NO6	248	112	*n/a*	*n/a*		*n/a*	*n/a*		−133.95	−117.11		*n/a*
3ON7	280	135	*n/a*	*n/a*		*n/a*	*n/a*		−116.88	−96.64		*-n/a*

n/a denotes the experimental results are not available.

Table 5
The average energy and average RMSD values achieved from two different variants of GA. The average values are calculated over 50 different runs. The bold-faced values indicate the winner (the lower the better).

Protein details			The state-of-the-art GA (Torres et al., 2007)				The MH_GA					
			Reported values				Average values					
			MJ model				MJ model		MH model			Gen
Seq	*Size*	*H*	*Energy*	*RMSD*	*Pop*	*Gen*	*Energy*	*RMSD*	*Energy*	*RMSD*	*Pop*	(≤)
2J6A	135	71	−815.82[a]	16.75	50	20,000	−59.72	9.53	**−61.40**	**9.48**	50	**2500**
2HFQ	85	38	−543.17[a]	12.24	50	20,000	−52.13	7.48	**−52.72**	**7.31**	50	**7000**

[a] The unusual values for MJ energy model.

Fig. 8. The performance hierarchy among the state-of-the-art approaches and our MH_GA. Our GA outperforms the other to approaches in Ullah and Steinhöfel (2010) and Shatabda et al. (2013a).

operator that repeatedly applies diagonal moves towards forming a hydrophobic core.
2. *MJ:* This version of our GA uses the MJ energy model for search and energy reporting. This version has macro-mutation operator but not biased by hydrophobic properties of amino acids.
3. *HP:* This version of our GA uses the HP energy model for search. However, we report the energy values of the final conformations returned by the GA in MJ energy model. Note that this version has the hydrophobic core directed macro-mutation operator. This version will show whether HP model is sufficient even when the energy of a conformation is to be in the MJ model.

From the Column RI in Table 8, we see that MH guided GA improves the average interaction energy level over MJ model, ranging from 0.84% to 5.14% for all benchmark proteins. The improvements are not large in magnitudes but consistently better for all the proteins.

5.3.2. Statistical significance

We know that the lower *p*-values are better. We performed the *t*-test with a confidence interval of 95% (i.e., significance level is 5%) and the results are presented in Table 8. For MJ and MH models, the *p*-values of all proteins are less than the significance level. However, for HP model, the *p*-value for 3MSE is below the significance level and for other five sequences those are equal to the significance level. Therefore, the experimental results are statistically significant.

5.3.3. Search progress

To demonstrate the search progress, we periodically find the best energy values obtained so far in each run. For a given period, we then calculate the average energy values obtained for that period over 50 runs. We used a 2-min time interval. Fig. 9 presents the average energy values obtained at each time interval for two different proteins: *4RXN* and *3PNX* are the smallest and largest amongst the 12 benchmark proteins respectively. From both of the charts, we see that the final version of our algorithm MH performs better than the other two versions.

6. Discussion

By encoding the conformation with angular coordinates (ϕ and ψ), our GA might easily be applied in high-resolution PSP. While the minimizing energy function is highly complex (such as molecular dynamics), a simple guidance heuristic—such as hydrophobic property or exposed surface area—could be used to guide the macro-mutation operator. Within GA framework, the macro-mutation operator could be applied optimizing the segments of secondary structures (α-helix and β-sheet).

Table 6
The best RMSD values reported, are the best amongst the 50 different runs. The bold-faced values indicate the winner (the lower the better).

Protein details			Local search (Shatabda et al., 2013a)	The MH_GA		
Seq	*Size*	*H*	*MJ guided*	*HP guided*	*MJ guided*	*MH guided*
4RXN	54	27	5.74	**4.70**	4.83	4.76
1ENH	54	19	5.94	**4.42**	4.75	4.81
4PTI	58	32	**6.02**	6.18	6.24	6.06
2IGD	61	25	7.38	7.64	6.63	**6.53**
1YPA	64	38	6.54	**5.17**	5.52	5.39
1R69	69	30	6.12	**4.44**	4.76	4.64
1CTF	74	42	6.08	4.72	4.26	**4.08**
3MX7	90	44	8.17	**7.10**	7.21	7.20
3NBM	108	56	6.38	5.89	5.64	**5.37**
3MQO	120	68	6.92	6.44	6.33	**6.38**
3MRO	142	63	8.76	7.76	7.93	**7.64**
3PNX	160	84	8.78	7.90	8.04	**7.60**
3MSE	180	83	*n/a*	20.24	**16.05**	16.98
3MR7	189	88	*n/a*	10.43	9.42	**9.36**
3MQZ	215	115	*n/a*	11.21	**8.88**	9.04
3NO3	238	102	*n/a*	14.49	**11.22**	11.70
3NO6	248	112	*n/a*	13.20	13.88	**12.04**
3ON7	280	135	*n/a*	13.19	11.84	**11.77**

Table 7
Corresponding MJ energies for global minimum RMSD and corresponding RMSDs for global minimum MJ energies over 50 runs for each proteins.

Protein details			Energy corresponds to RMSD						RMSD corresponds to energy					
			HP		MJ		MH		HP		MJ		MH	
Seq	*Size*	*H*	*rmsd*	*En*	*rmsd*	*En*	*rmsd*	*En*	*En*	*rmsd*	*En*	*rmsd*	*En*	*rmsd*
4RXN	54	27	4.70	4.24	4.83	−26.68	4.76	−26.02	−12.41	6.30	−37.06	5.91	−36.36	5.99
1ENH	54	19	4.42	−0.67	4.75	−15.21	4.81	−10.8	−10.27	7.26	−38.85	7.68	−38.39	7.14
4PTI	58	32	6.18	−0.36	6.24	−8.03	6.06	−19.16	−6.95	7.00	−32.6	8.09	−35.65	8.62
2IGD	61	25	7.64	4.00	6.63	−18.21	6.53	−19.79	−10.28	9.4	−35.57	9.86	−36.49	8.69
1YPA	64	38	5.17	5.21	5.52	−26.90	5.39	−35.01	−17.1	8.37	−38.45	7.81	−40.14	8.32
1R69	69	30	4.44	3.59	4.76	−21.70	4.64	−22.37	−11.3	5.38	−39.89	7.16	−40.85	6.40
1CTF	74	42	4.72	−3.72	4.26	−32.55	4.08	−44.44	−18.06	7.19	−50.45	5.96	−51.50	5.94
3MX7	90	44	7.10	−0.08	7.21	−42.18	7.20	−50.85	−17.97	8.73	−56.55	10.05	−56.32	9.57
3NBM	108	56	5.89	−5.07	5.64	−35.75	5.37	−36.51	−23.09	8.27	−55.38	6.75	−53.66	7.34
3MQO	120	68	6.44	5.96	6.33	−51.44	6.38	−41.69	−15.47	9.31	−62.65	7.69	−62.25	8.13
3MRO	142	63	7.76	−10.97	7.93	−50.69	7.64	−68.41	−28.63	12.96	−90.56	11.89	−90.05	9.28
3PNX	160	84	7.90	−1.16	8.04	−73.90	7.60	−69.52	−26.79	10.81	−96.98	10.11	−102.55	10.12
3MSE	180	83	20.24	−14.41	16.05	−76.99	16.98	−77.73	−30.4	22.01	−91.02	19.12	−92.61	17.88
3MR7	189	88	10.43	−12.34	9.42	−84.28	9.36	−81.9	−26.99	10.56	−94.93	11.67	−93.65	10.84
3MQZ	215	115	11.21	−5.26	8.88	−98.75	9.04	−92.85	−15.51	11.53	−108.38	10.58	−104.29	10.7
3NO3	238	102	14.49	−14.51	11.22	−112.14	11.7	−100.79	−16.41	14.89	−119.9	13.2	−122.97	13.04
3NO6	248	112	13.2	−8.67	11.88	−120.23	12.06	−116.51	−44.07	13.96	−125.68	14.26	−133.95	13.09
3ON7	280	135	13.19	28.47	11.84	−105.63	11.77	−98.31	−8.59	13.95	−120.16	13.01	−116.88	16.58

Table 8
The effect of using HP energy model within a macro-mutation operator. The bold-faced values indicate the winners. The lower the energy value, the better the performance. The *t*-test was performed with a confidence interval of 95%.

Protein details			Best of 50 runs			Average [*p*-value] of 50 runs			RI
Seq	*Size*	*H*	*HP*	*MJ*	*MH*	*HP*	*MJ* (*r*)	*MH* (*t*)	*on MJ*
4RXN	54	27	−12.41	**−37.71**	−36.36	−3.54 [2.4E−16]	−33.32 [5.9E−56]	**−33.60** [1.7E−75]	0.84%
1ENH	54	19	−10.27	−37.37	**−38.39**	−7.29 [3.8E−32]	−34.86 [1.1E−66]	**−35.67** [1.2E−70]	2.32%
4PTI	58	32	−6.95	−35.31	**−35.65**	−2.81 [1.5E−14]	−30.93 [3.6E−55]	**−31.01** [4.8E−67]	0.26%
2IGD	61	25	−10.28	**−36.97**	−36.49	−6.75 [2.7E−31]	−33.65 [3.5E−66]	**−33.75** [4.0E−70]	0.30%
1YPA	64	38	−17.1	−39.13	**−40.14**	−9.90 [2.3E−33]	−35.20 [6.4E−65]	**−36.33** [2.8E−73]	3.21%
1R69	69	30	−11.3	−39.77	**−40.85**	−4.31 [5.6E−19]	−35.43 [4.9E−65]	**−36.28** [2.5E−68]	2.40%
1CTF	74	42	−18.06	−50.09	**−51.5**	−10.97 [1.1E−32]	−44.98 [1.4E−61]	**−47.29** [6.8E−70]	5.14%
3MX7	90	44	−17.97	−55.57	**−56.32**	−11.16 [1.9E−31]	−48.46 [5.5E−62]	**−50.95** [2.6E−70]	5.14%
3NBM	108	56	−23.09	**−57.17**	−53.66	−15.29 [9.8E−36]	−48.47 [9.5E−60]	**−49.90** [2.6E−70]	2.95%
3MQO	120	68	−15.47	−60.22	**−62.25**	−6.75 [1.7E−18]	−53.00 [4.8E−61]	**−54.56** [2.4E−66]	2.94%
3MRO	142	63	−28.63	**−93.77**	−90.05	−18.65 [7.2E−31]	−79.32 [2.1E−62]	**−82.32** [1.6E−67]	3.78%
3PNX	160	84	−26.79	−99.87	**−102.55**	−18.55 [1.2E−34]	−85.64 [6.0E−60]	**−88.06** [1.3E−60]	2.83%
3MSE	180	83	−30.4	−91.02	−92.61	−13.17 [5.0E−21]	−84.47 [3.2E−70]	**−84.60** [3.6E−69	0.20%
3MR7	189	88	−26.99	−94.93	−93.65	−5.54 [1.4E−06]	**−85.70** [4.1E−69]	−83.93 [1.9E−36]	*non*
3MQZ	215	115	−15.51	−108.38	−104.29	6.86 [8.7E−08]	**−96.58** [1.5E−68]	−95.22 [6.7E−64]	*non*
3NO3	238	102	−16.41	−119.9	−122.97	−2.41 [5.1E−02]	−108.68 [1.1E−68]	**−108.70** [3.3E−65]	0.12%
3NO6	248	112	−44.07	−125.68	−133.95	−12.65 [2.0E−11]	−116.31 [1.8E−71]	**−117.11** [7.0E−67]	0.70%
3ON7	280	135	−8.59	−120.16	−116.88	9.38 [7.0E−10]	**−104.57** [1.1E−56]	−96.64 [2.4E−45]	*non*

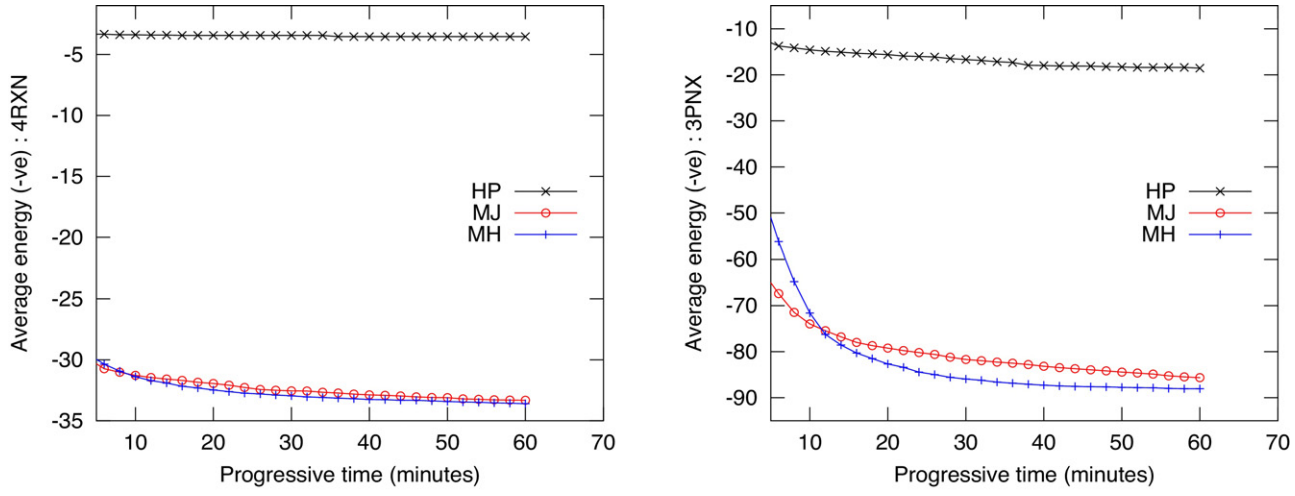

Fig. 9. The search progress over a time-span of 60 min for proteins *4RXN* and *3PNX* of sequence length 54 and 160 amino acids respectively.

Our approach can easily divide the whole optimization process into two stages guided by two energy models with different complexities. The macro-mutation operator can be guided by simpler energy models such as distance from hydrophobic core, exposed surface area, hydrophobicity of amino acids, hydropathy index of the amino acids, and so on. Conversely, the main objective function can be more realistic such as molecular dynamics based energy models. This two-stage optimization will reduce the overall computational complexities. As a result, our framework has a good chance to succeed in more realistic models even for large sized proteins.

7. Conclusion

Our guided macro-mutation in a graded energy based genetic algorithm, 'MH GeneticAlgorithm', is found to be an effective sampling algorithm for the convoluted protein structure space. The strategical switching in between the Miyazawa–Jernigan (MJ) energy and the hydrophobic-polar (HP) energy made the proposed algorithm perform better compared to the other state-of-the-art approaches. This is because, while the fine graded MJ energy interaction computation become computationally prohibit, the low resolution HP energy model can effectively sample the search-space towards certain promising directions. In addition, the GA framework was enhanced and made powerful, since it uses not only crossover but also three effective move operators. Further to diversify the population to keep sampling or, exploring the search space effectively, a hydrophobic core-directed macro-mutation operator, twin removal as well as a random-walk algorithm to recover from the stagnation has been applied. To compare the performance of our GA, we have extensively compared with the existing state-of-the-approaches using the available benchmark problems and found our approach to be consistently better as well as often found significantly better – t-test result in terms of p-values have been provided to support the claims. For the lattice configuration to be followed, we used 3D face-centered-cube (FCC) lattice model, because prediction in the FCC lattice model can yield the densest protein core and the FCC lattice model can provide the maximum degree of freedom as well as the closest resemblance to the real or, high resolution folding within the lattice constraint. This enables the predicted structure to be aligned and hence, migrated to a real protein (prediction) model efficiently for future extensions.

Acknowledgments

Mahmood Rashid and Abdul Sattar would like to express their great appreciation to National ICT Australia. Sumaiya Iqbal and Md Tamjidul Hoque acknowledge the Louisiana Board of Regents through the Board of Regents Support Fund, LEQSF(2013-16)-RD-A-19.

References

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walters, P., 2002. The shape and structure of proteins. In: Mol. Bio. of the Cell, fourth ed.
Alm, E., Baker, D., 1999. Matching theory and experiment in protein folding. Curr. Opin. Struct. Biol. 9 (2), 189–196.
Anfinsen, C.B., 1973. The principles that govern the folding of protein chains. Science 181 (4096), 223–230.
Böckenhauer, H.-J., Ullah, A.Z.M.D., Kapsokalivas, L., Steinhöfel, K.,2008. A local move set for protein folding in triangular lattice models. In: WABI, vol. 5251 of Lecture Notes in Computer Science. Springer, pp. 369–381.
Bahi, J.M., Guyeux, C., Mazouzi, K., Philippe, L., 2013. Computational investigations of folded self-avoiding walks related to protein folding. Comput. Biol. Chem. 47, 246–256.
Baker, D., 2000. A surprising simplicity to protein folding. Nature 405 (6782), 39–42.
Berrera, M., Molinari, H., Fogolari, F., 2003. Amino acid empirical contact energy definitions for fold recognition in the space of contact maps. BMC Bioinform. 4 (1), 8.
Blum, C., 2005. Ant colony optimization: introduction and recent trends. Phys. Life Rev. 2 (4), 353–373.
Cebrián, M., Dotú, I., Van Hentenryck, P., Clote, P., 2008. Protein structure prediction on the face centered cubic lattice by local search. In: Proceedings of the 23rd national conference on Artificial intelligence, vol. 1, pp. 241–246.
Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popoviă, Z., Players, F., 2010. Predicting protein structures with a multiplayer online game. Nature 466 (7307), 756–760.
Cutello, V., Nicosia, G., Pavone, M., Timmis, J., 2007. An immune algorithm for protein structure prediction on lattice models. IEEE Trans. Evol. Comput. 11 (1), 101–117.
Dal Palù, A., Dovier, A., Fogolari, F., 2004. Constraint logic programming approach to protein structure prediction. BMC Bioinform. 5 (1), 186.
Dal Palù, A., Dovier, A., Pontelli, E., 2005. Heuristics, optimizations, and parallelism for protein structure prediction in clp (fd). In: Proceedings of the 7th International Conference on Principles and Practice of Declarative Programming, pp. 230–241.
Dal Palù, A., Pontelli, E., Dovier, A., 2007. A constraint solver for discrete lattices, its parallelization, and application to protein structure prediction. Softw. Pract. Exp. 37 (13), 1405.
Dal Palù, A., Dovier, A., Fogolari, F., Pontelli, E., 2011. Exploring protein fragment assembly using CLP. In: Proceedings of the International Joint Conference on Artificial Intelligence 3, pp. 2590–2595.
Das, R., Baker, D., 2008. Macromolecular modeling with rosetta. Annu. Rev. Biochem. 77, 363–382.
Dobson, C.M., 2003. Protein folding and misfolding. Nature 426 (6968), 884–890.
Dodson, Eleanor J., 2007. Computational biology: protein predictions. Nature 450 (7167), 176–177.
Dotú, I., Cebrián, M., Van Hentenryck, P., Clote, P., 2011. On lattice protein structure prediction revisited. IEEE Trans. Comput. Biol. Bioinform. 8 (6), 1620–1632.
Giaquinta, E., Pozzi, L., 2013. An effective exact algorithm and a new upper bound for the number of contacts in the hydrophobic-polar two-dimensional lattice model. J. Comput. Biol. 20 (8), 593–609.
Hales, T.C., 2005. A proof of the Kepler conjecture. Ann. Math. 162 (3), 1065–1185.
Higgs, T.B., Stantic, B., Hoque, M.T., Sattar, A., 2012a. Applying Feature-based Resampling to Protein Structure Prediction. http://cs.uno.edu/tamjid/Papers/2012_FBR_PSP.pdf.
Higgs, T., Stantic, B., Hoque, M.T., Sattar, A.,2012b. Refining genetic algorithm twin removal for high-resolution protein structure prediction. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–8.
Holland, J.H., 1975. Adaptation in Natural and Artificial System: An Introduction with Application to Biology, Control and Artificial Intelligence. University of Michigan Press, Ann Arbor.
Hoque, M.T., 2007, September. Genetic Algorithm for *Ab initio* Protein Structure Prediction based on Low Resolution Models (Ph.D. thesis). Monash University, Australia.
Hoque, M.T., Chetty, M., Dooley, L.S., 2005. A new guided genetic algorithm for 2D hydrophobic–hydrophilic model to predict protein folding. In: IEEE Congress on Evolutionary Computation, vol. 1, pp. 259–266.
Hoque, M.T., Chetty, M., Dooley, L.S.,2006. Non-isomorphic coding in lattice model and its impact for protein folding prediction using genetic algorithm. In: IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology. IEEE, pp. 1–8.
Hoque, M.T., Chetty, M., Sattar, A., 2007a. Protein folding prediction in 3D FCC HP lattice model using genetic algorithm. In: CEC 2007. IEEE Congress on Evolutionary Computation, pp. 4138–4145.
Hoque, M.T., Chetty, M., Dooley, L.S.,2007b. Generalized schemata theorem incorporating twin removal for protein structure prediction. In: Pattern Recognition in Bioinformatics. Springer, pp. 84–97.
Hoque, M.T., Chetty, M., Lewis, A., Sattar, A., Avery, V.M., 2010. DFS-generated pathways in ga crossover for protein structure prediction. Neurocomputing 73 (13), 2308–2316.
Hoque, M.T., Chetty, M., Lewis, A., Sattar, A., 2011. Twin removal in genetic algorithms for protein structure prediction using low-resolution model. IEEE/ACM Trans. Comput. Biol. Bioinform. 8 (1), 234–245.
Iqbal, S., Mishra, A., Hoque, M.T., 2015. Improved prediction of accessible surface area results in efficient energy function application. J. Theor. Biol. 380, 380–391.
Islam, M.K., 2011. Memetic Approach for Prediction of Low Resolution Protein Structures using Lattice Models (Ph.D. thesis). Monash University, Victoria, Australia.
Islam, M.K., Chetty, M., 2009. Novel memetic algorithm for protein structure prediction. In: Australasian Conference on Artificial Intelligence, pp. 412–421.
Islam, M.K., Chetty, M., 2010. Clustered memetic algorithm for protein structure prediction. In: IEEE Congress on Evolutionary Computation, pp. 1–8.
Islam, M.K., Chetty, M., 2013. Clustered memetic algorithm with local heuristics for ab initio protein structure prediction. IEEE Trans. Evol. Comput. 17 (4), 558–576.
Islam, M.K., Chetty, M., Murshed, M., 2011a. Conflict resolution based global search operators for long protein structures prediction. In: ICONIP (1), pp. 636–645.
Islam, M.K., Chetty, M., Murshed, M., 2011b. Novel local improvement techniques in clustered memetic algorithm for protein structure prediction. In: IEEE Congress on Evolutionary Computation, pp. 1003–1011.
Islam, M.K., Chetty, M., Ullah, A.Z.M.D., Steinhöfel, K., 2011c. A memetic approach to protein structure prediction in triangular lattices. In: ICONIP (1), pp. 625–635.
Istrail, S., Lam, F., et al., 2009. Combinatorial algorithms for protein folding in lattice models: a survey of mathematical results. Commun. Inform. Syst. 9 (4), 303–346.
Kapsokalivas, L., Gan, X., Albrecht, A.A., Steinhöfel, K., 2009. Population-based local search for protein folding simulation in the MJ energy model and cubic lattices. Comput. Biol. Chem. 33 (4), 283–294.

Kaufmann, K.W., Lemmon, G.H., DeLuca, S.L., Sheehan, J.H., Meiler, J., 2010. Practically useful: what the Rosetta protein modeling suite can do for you. Biochemistry 49 (14), 2987–2998.
Kern, C., Liao, L., 2013. Lattice models with asymmetric propensity matrices for locationally informed protein structure prediction. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 90–93.
Klau, G.W., Lesh, N., Marks, J., Mitzenmacher, M., 2002. Human-guided tabu search. In: The Eighteenth National Conference on Artificial Intelligence (AAAI-02).
Kondov, I., Berlich, R., 2011. Protein structure prediction using particle swarm optimization and a distributed parallel approach. In: ACM Workshop on Biologically Inspired Algorithms for Distributed Systems, BADS '11, pp. 35–42.
Krasnogor, N., Blackburne, B.P., Burke, E.K., Hirst, J., 2002. Multimeme algorithms for protein structure prediction. In: Parallel Problem Solving from Nature – PPSN VII, vol. 2439, pp. 769–778.
Lau, K.F., Dill, K.A., 1989. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules 22 (10), 3986–3997.
Lee, J., Wu, S., Zhang, Y.,2009. Ab initio protein structure prediction. In: From Protein Structure to Function with Bioinformatics. Springer, pp. 3–25.
Lesh, N., Mitzenmacher, M., Whitesides, S.,2003. A complete and effective move set for simplified protein folding. In: Research in Computational Molecular Biology. ACM, pp. 188–195.
Levinthal, C., 1968. Are there pathways for protein folding? J. Med. Phys. 65 (1), 44–45.
Maher, B., Albrecht, A.A., Loomes, M., Yang, X.-S., Steinhöfel, K., 2014. A firefly-inspired method for protein structure prediction in lattice models. Biomolecules 4 (1), 56–75.
Mann, M., Will, S., Backofen, R., 2008. CPSP-tools – exact and complete algorithms for high-throughput 3D lattice protein studies. BMC Bioinform. 9 (1), 230.
Mansour, N., Kanj, F., Khachfe, H., 2012. Particle swarm optimization approach for protein structure prediction in the 3D HP model. Interdiscip. Sci. 4 (3), 190–200.
Miyazawa, S., Jernigan, R.L., 1985. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. Macromolecules 18 (3), 534–552.
Morowitz, H.J., 1968. Energy Flow in Biology. Academic Press.
Pelta, D.A., Krasnogor, N.,2005. Multimeme algorithms using fuzzy logic based memes for protein structure prediction. In: Recent Advances in Memetic Algorithms. Springer, pp. 49–64.
Rashid, M.A., Hoque, M.T., Newton, M.A.H., Pham, D., Sattar, A.,2012a. A new genetic algorithm for simplified protein structure prediction. In: Advances in Artificial Intelligence. Springer, Berlin, Heidelberg, pp. 107–119.
Rashid, M.A., Shatabda, S., Newton, M.A.H., Hoque, M.T., Pham, D.N., Sattar, A.,2012b. Random-walk: a stagnation recovery technique for simplified protein structure prediction. In: BCB. ACM, pp. 620–622.
Rashid, M.A., Newton, M.A.H., Hoque, M.T., Sattar, A.,2013a. A local search embedded genetic algorithm for simplified protein structure prediction. In: IEEE Congress on Evolutionary Computation. IEEE, pp. 1091–1098.
Rashid, M.A., Hoque, M.T., Newton, M.A.H., Sattar, A.,2013b. Collaborative parallel local search for simplified protein structure prediction. In: 12th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA. IEEE Computer Society, pp. 966–973.
Rashid, M.A., Newton, M.A.H., Hoque, M.T., Shatabda, S., Pham, D., Sattar, A., 2013c. Spiral search: a hydrophobic-core directed local search for simplified PSP on 3D FCC lattice. BMC Bioinform. 14 (Suppl 2), S16.
Rashid, M.A., Newton, M.A.H., Hoque, M.T., Sattar, A., 2013d. Mixing energy models in genetic algorithms for on-lattice protein structure prediction. BioMed Res. Int., 15.
Rashid, M.A., Khatib, F., Hoque, M.T., Sattar, A., 2015. An enhanced genetic algorithm for *Ab initio* protein structure prediction. IEEE Trans. Evol. Comput., http://dx.doi.org/10.1109/TEVC.2015.2505317.
Shatabda, S., Newton, M.A.H., Pham, D.N., Sattar, A.,2012. Memory-based local search for simplified protein structure prediction. In: The ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM-BCB). ACM, Orlando, FL, USA.
Shatabda, S., Newton, M.A.H., Sattar, A., 2013a. Mixed heuristic local search for protein structure prediction. In: Proceedings of the Twenty-seventh AAAI Conference on Artificial Intelligence.
Shatabda, S., Newton, M.A.H., Rashid, M.A., Sattar, A., 2013b. An efficient encoding for simplified protein structure prediction using genetic algorithms. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 1217–1224.
Smith, Adam, 2003. Protein misfolding. Nat. Rev. Drug Discov. 426 (6968), 78–102.
Stouthamer, A., 1973. A theoretical study on the amount of ATP required for synthesis of microbial cell material. Antonie van Leeuwenhoek 39 (1), 545–565.
Tantar, A.-A., Melab, N., Talbi, E.-G., 2008. A grid-based genetic algorithm combined with an adaptive simulated annealing for protein structure prediction. Soft Comput. 12 (12), 1185–1198.
Thachuk, C., Shmygelska, A., Hoos, H.H., 2007. A replica exchange Monte Carlo algorithm for protein folding in the HP model. BMC Bioinform. 8 (1), 342.
Torres, S.R.D., Romero, D.C.B., Vasquez, L.F.N., 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. ACM, pp. 393–400.
Ullah, A.D., Steinhöfel, K., 2010. A hybrid approach to protein folding problem integrating constraint programming with local search. BMC Bioinform. 11 (Suppl. 1), S39.
Ullah, A.D., Kapsokalivas, L., Mann, M., Steinhöfel, K.,2009. Protein folding simulation by two-stage optimization. In: Computational Intelligence and Intelligent Systems, vol. 51. Springer, Berlin, Heidelberg, pp. 138–145.
Unger, R., Moult, J., 1993a. Genetic algorithms for protein folding simulations. J. Mol. Biol. 231 (1), 75–81.
Unger, R., Moult, J.,1993b. A genetic algorithm for 3D protein folding simulations. In: The 5th International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, p. 581.
Wroe, R., Bornberg-Bauer, E., Chan, H.S., 2005. Comparing folding codes in simple heteropolymer models of protein evolutionary landscape: robustness of the superfunnel paradigm. Biophys. J. 88 (1), 118–131.
Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. Proteins 80 (7), 1715–1735.
Yue, K., Dill, K.A., 1993. Sequence-structure relationships in proteins and copolymers. Phys. Rev. E 48 (3), 2267.