Protein Structure Prediction on a Lattice Model via Multimodal Optimization Techniques

Ka-Chun Wong, Kwong-Sak Leung, Man-Hon Wong
Department of Computer Science & Engineering
The Chinese University of Hong Kong, HKSAR, China
{kcwong, ksleung, mhwong}@cse.cuhk.edu.hk

ABSTRACT

This paper considers the protein structure prediction problem as a multimodal optimization problem. In particular, de novo protein structure prediction problems on the 3D Hydrophobic-Polar (HP) lattice model are tackled by evolutionary algorithms using multimodal optimization techniques. In addition, a new mutation approach and performance metric are proposed for the problem. The experimental results indicate that the proposed algorithms are more effective than the state-of-the-arts algorithms, even though they are simple.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Measurement

Keywords

Protein Structure Prediction, Multimodal Optimization, HP Lattice Model, Relative Encoding, Absolute Encoding, Distance Metric, Crowding, Fitness Sharing, Evolutionary Algorithm

1. INTRODUCTION

A polypeptide is a chain of amino acid residues. Once folded into its native state, it is called a protein. Proteins plays vital roles in living organisms. They perform different tasks to maintain a body's life. For instance, material transportations across cells, catalyzing metabolic reactions and body defenses against viruses. Nevertheless, the functions of proteins substantially depend on their structural features. In other words, researchers need to know a protein's native structure before its function can be completely deduced. It gives rises to the protein structure prediction problem.

The protein structure prediction problem is often referred as the "holy grail" of biology. In particular, Anfinsen's dogma [2] and Levinthal's paradox [21] play an important role in the problem. Anfinsen's dogma postulates that the native structure of a protein (tertiary structure) only depends on its amino acid residue sequence (primary structure). On the other hand, Levinthal's paradox postulates that it is too time-consuming for a protein to randomly sample all the feasible confirmation regions for its native structure. But, on the other hand, the proteins in nature can still spontaneously fold into their native structures in about several milliseconds.

Based on the above ideas, researchers have explored the problem throughout several years. In particular, the designability of a structure and the degeneracy of a sequence have been studied by Li et. al. [22]. The computational complexity has also been examined by Hart et. al. [1].

Numerous prediction approaches have been proposed. In general, they can be classified into two categories, depending on whether any prior knowledge other than sequence data has been incorporated [4]. This paper focuses on De novo (or Ab initio) protein structure prediction on the 3D Hydrophobic-Polar (HP) lattice model using evolutionary algorithms [20]. In other words, only sequence data is considered.

2. BACKGROUND

2.1 HP Lattice Model

2.1.1 Motivation

Different protein structure models have been proposed in the past [24]. Their differences mainly lies in their resolution levels and search space freedom. For the highest resolution levels, all the atoms and bond angles can be simulated using molecular dynamics. Nevertheless, there is no free lunch. The simulation is hard to be completed by the current computational power. On the other hand, a study indicated that protein folding mechanisms might be simpler than the previous thought [3]. Simplified models are enough. Thus this paper focuses on the HP lattice model to capture the physical principles of the protein folding process [11, 27].

2.1.2 Description

HP lattice model was proposed by Dill [10]. It assumes that the main driving forces are the interactions among the hydrophobic amino acid residues. The twenty types of amino acids are experimentally classified as either hydrophobic (H)

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

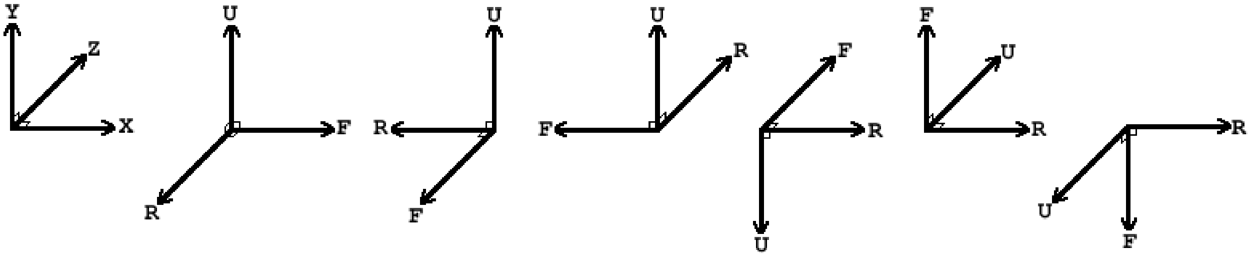


Figure 1: Relative Encoding used in this paper

or polar (P). An amino acid residue sequence is thus represented as a string $\{H, P\}^+$. Each residue is represented as a non-overlapping bead in a cubic lattice. Each peptide bond in the main chain is represented as a connecting line. A protein is thus represented as a non-overlapping chain in the cubic lattice.

2.1.3 Objective

Based on the above model, the objective of the protein structure prediction problem is to find the conformation with the minimal energy for each protein. Mathematically, it is to minimize the following function [22]:

$$H = \sum_{i+1<j} E_{\sigma_i \sigma_j} \Delta(r_i - r_j)$$

where r_i and r_j are the amino acid residues at the sequence position i and j. The constraint $i + 1 < j$ is to ensure that r_i and r_j are not next to each other in their sequence and examined together once only. $\Delta(r_i - r_j) = 1$ when r_i and r_j are adjacent in the lattice space, otherwise $\Delta(r_i - r_j) = 0$. As stated in the previous section, each residue is represented as either H or P. Thus $E_{\sigma_i \sigma_j}$ could be E_{HH}, E_{HP}, E_{PH}, and E_{PP}. For their values, three schemes have been proposed. The most widely used scheme is $E_{HH} = -1$, $E_{HP} = 0$, $E_{PH} = 0$, and $E_{PP} = 0$. The second scheme $E_{HH} = -2.3$, $E_{HP} = -1$, $E_{PH} = -1$, and $E_{PP} = 0$ was proposed by [22]. The last scheme $E_{HH} = -2$, $E_{HP} = 1$, $E_{PH} = 1$, and $E_{PP} = 1$ is called functional model protein (or "shifted" HP model) [9]. As mentioned in [24], the results are insensitive to the value of E_{HH} as long as the physical constraints [22] are satisfied. Thus we have chosen the first scheme in the following sections.

2.2 Representation

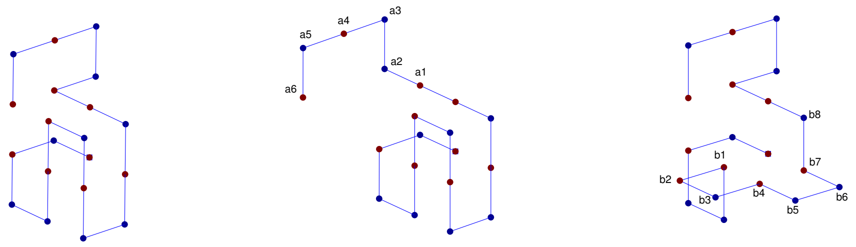
For the representation of an amino acid residue sequence, there are two conditions to be satisfied: [20]

1. Sequence connectivity
2. Self-avoidance

Among the representations proposed [9], *Internal Coordinate* should be a favorable choice since it can handle the first condition implicitly. Internal Coordinate is a representation system which residue positions depend on their sequence-predecessor residues. There are two types of Internal Coordinate representation: *Absolute Encoding* and *Relative Encoding*. Absolute Encoding represents each residue position as the absolute direction from the previous residue. A sequence is represented as $\{U, D, L, R, F, B\}^{n-1}$ (Up, Down, Left, Right, Forward, Backward) [29]. On the other hand, Relative Encoding represents each residue position as the direction relative to the previous direction of the two predecessor residues. Backward direction is omitted for one-step self-avoiding. Thus a sequence is represented as $\{F, R, L, U, D\}^{n-2}$ [26]. Except the forward move, a cyclic conformation is formed if a move is repeated four times. Krasnogor et al. [20] have examined both representations on square lattices. Their results showed that Relative Encoding had better performance than Absolute Encoding on square lattices. Our preliminary results also indicated that the performance of Absolute Encoding degraded as a sequence got longer on cubic lattices. Thus we have chosen Relative Encoding as the representation in the following sections. For this representation, different orientations can be taken. Nevertheless, few explicitly stated their representations in a pictorial way. Thus the representation we have adopted is depicted in Fig.1 for the sake of clarity. The most left sub-figure denotes the absolute direction axis, whereas the remaining sub-figures denotes the Relative Encoding representations for all the six directions in cubic lattices. For instance, the second left sub-figure denotes the Relative Encoding representation the subsequent move should use when the current move is in the positive X direction. In particular, the subsequent move is called a forward move if it is still in the positive X direction.

2.3 Related Works

Although the 3D HP model seems relatively simple among other models, it has been proved that the protein structure prediction problem on the model is NP-Complete [5]. Thus researchers propose heuristics as compromising solutions. In particular, the seminal work by Unger et al. [29] experimentally showed that genetic algorithm approaches were better than Monte Carlos simulations. Thus many researchers tried genetic algorithm as one of the heuristics to solve the problem. Nevertheless, the genetic algorithm approach by Unger et al. [29] was actually hybridized with Monte Carlo moves. Hence Patton et al. [26] further generalized it into a standard genetic algorithm approach, which search space included infeasible regions penalized by a penalty function. Furthermore, they proposed Relative Encoding so that one-step self-avoiding constraints could be implicitly incorporated in the genome representation. Few years later, Krasnogor et al. [20] published a work discussing the basic algorithmic factors affecting the problem. Since then, researchers explored different ways to tackle the problem. For instance, Krasnogor et al. further applied a multimeme algorithm, which adaptively chose multiple local searchers to reach optimal structures [19]. Cox et al. [8] and Hoque et al. [15] utilized heavy machinery of specific genetic operators and techniques. Ant colony algorithm [28], differential evolution



(a) An optimal conformation (b) Example A (c) Example B

Figure 2: Some conformations of UM20 [7]

[6], immune algorithm [9] and estimation of distribution algorithm [27] were also customized and reported in literatures. In particular, some diversity preserving techniques were often incorporated in them. For instance, duplicate predator [8], aging operator [9], and additional renormalization of the pheromone [28]. They can be deemed as the signs of the multimodality in the problem. However, to the authors' knowledge, none of them has explicitly focused on the necessity of multimodal optimization techniques.

3. MULTIMODAL OPTIMIZATION

3.1 Motivation

For the protein structure prediction problem, it is generally believed that the native state of a protein should be in the conformation with the lowest energy. Thus the previous works mainly focus on the minimal energy they could achieve: the minimal energy ever found ($H(x)$) and the average and standard deviation of the minimal energy across several runs ($mean \pm \sigma$).

Nevertheless, Jahn et al. [16] has shown that the native state is not necessarily a single global optimum. It may also be a local optimum in Fig.1 of [16]. For the HP lattice model, Unger et al. [30] have observed that there can be multiple conformations for each energy value. A recent fitness landscape study also indicated that HP landscapes were multimodal [12].

Thus we propose applying multimodal optimization techniques to the problem explicitly in this paper, in order to preserve diversity. In other words, building blocks and optima can be preserved. A more effective search is guaranteed throughout each run. Both global and local optima are more likely to be found. The native state information is less likely to be lost.

3.2 Multimodal Optimization Techniques

The work by De Jong [17] is the first known attempt to solve multimodal optimization problems. He introduced the crowding technique to increase the chance for locating multiple optima: an offspring can only replace the parent which is most similar to the offspring itself. Such a strategy can preserve the diversity and maintain different types of individuals in a run. Twelve years later, Goldberg et al. [13] proposed a fitness-sharing niching technique as a diversity preserving strategy. He proposed a shared fitness function, instead of an absolute fitness function, to evaluate the fitness of an individual in order to favor the growth of the individuals which are distinct to others. With this technique, a population can be prevented from the domination of a particular type of individuals. Species conserving genetic algorithm(SCGA) [23] is another technique for evolving parallel subpopulations. Before each generation starts to crossover, the algorithm selects a set of species seeds which can bypass the subsequent procedures to the next generation.

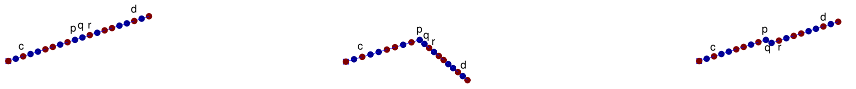
The previous techniques are the backbone techniques for multimodal optimization. All of them are implemented and tested for the protein structure problem in the following sections. Historically, they were originally designed for real number optimization. Careful modifications are needed before applying them to the protein structure prediction problem. In particular, there are two critical factors to be considered:

- How to determine the distance between two conformations? The most widely used distance measure should be the root mean square deviation (RMSD) [14]. RMSD calculates the average absolute distances between two superimposed conformations' points. Nevertheless, if two conformations differ by only one point direction in Relative Encoding, their RMSD cannot reflect such small change. For instance, some conformations of the benchmark UM20 [7] are visualized in Fig.2. Fig.2(a) depicts one of the optimal conformations. The other sub-figures depict two candidate conformations:

  – `Optimal   : LDLDFLUFDDFRFRDDFD`
  – `Example A: LDLDFLUFDDFRFFDDFD`
  – `Example B: LDLDDLLRLLDRFRDDFD`

  To be mutated to the optimal conformation, Example A is only needed to change its move between a1 and a2 to R whereas example b is needed to change nearly all of its moves between b1 and b8. However, the RMSD between Example A and the optimal conformation (5 diagonal point changes a2 to a6) is larger than that between example B and the optimal conformation (4 diagonal point changes b2,b3,b5,b6). RMSD cannot capture the move information in Relative Encoding.

  Furthermore, if RMSD is applied in our algorithms, it will be quite computationally intensive: To calculate the RMSD between two conformations, the corresponding Relative Encoding genomes are converted to absolute 3D coordinates. Once converted, one of them is then translated and rotated to be optimally superimposed on the other. RMSD is then calculated which



(a) Before a mutation (b) After a RM (c) After a AM

Figure 3: The conformations before and after a mutation

involves multiplications and square root calculations. In contrast, Hamming distance calculates the move differences between two Relative Encoding genomes. It is relatively computational tractable. Thus Hamming distance is adopted as the distance metric in this paper.

Using Hamming distance, rotational symmetry can be implicitly handled by omitting the first move in the Relative Encoding representation. That's why a confirmation is represented as $\{F, R, L, U, D\}^{n-2}$, instead of $\{F, R, L, U, D\}^{n-1}$ in the previous sections. In other words, a single sequence in Relative Encoding representation actually represents the confirmations for all the six rotational directions in cubic lattices. Nevertheless, as a trade-off, mirror symmetry has not been handled in the distance metric. In the future, improvements will be definitely needed. One of the possible solution may draw the inspirations from the contact map memory [19].

- How to handle infeasible conformations? Basically there are two approaches:

  – Delete infeasible conformations
  – Tolerate infeasible conformations by adjusting their energy values by a penalty score

  Both approaches were thought beneficial in different view angles [29, 24, 20, 12]. For the first approach, it is conjectured that search space can be smaller if infeasible conformations are deleted. For the second approach, it is conjectured that the paths to optimal conformations are shorter if infeasible conformations exist. Nevertheless, the study in [12] had a detailed analysis supporting the first approach. Furthermore, our problem is a discrete optimization problem. Unlike continuous optimization, its gene can easily flip between different values. Thus there may be alternative paths to optimal conformations even if infeasible conformations are disallowed. Thus the first approach is taken in this paper.

Having decided the distance and infeasible confirmation handling methods, the multimodal optimization techniques are applied to the protein structure prediction problem. In addition, a mixed mutation method is also proposed and examined.

3.3 Proposed Mutation

As discussed in [20], the mutations in Relative Encoding differ from those in Absolute Encoding. For instance, suppose we have a straight sequence as shown in Fig.3. The resultant conformations after the two mutations are shown in Fig.3(b) and Fig.3(c). Preceding the mutation point p, there is no changes (e.g. point c). The affected sequence regions always lie in the points (e.g. point d) succeeding the mutation point p. These points are rotated by ninety degrees during the mutation in Relative Encoding (RM), whereas these points are merely translated diagonally by one unit during the mutation in Absolute Encoding (AM). The degree of changes is higher in RM than AM. Thus RM and AM can be thought as a pair of coarse-adjusting and fine-tuning operations. A mixed use of them is motivated. Therefore, a straightforward approach is proposed as shown in Algorithm 1.

Without loss of generality, a simple threshold is chosen (0.8 in this paper). If a random number generator generates a value higher than the threshold, AM is taken. Otherwise, RM is taken. To fully model AM and RM as a pair of coarse-adjusting and fine-tuning operations, AM only mutates one gene whereas RM tries to mutate all the genes in this paper.

To implement AM in Relative Encoding, an approach is outlined in the procedure *AbsolutelyMutate* in Algorithm 1. Fig.3 depicts an example. Before the (forward) move between point p and q is mutated (Fig.3(a)), the absolute direction of the subsequent move between point q and r (positive X) is saved. Once saved, the (forward) move between point p and q is randomly mutated to another direction. This (right) direction is examined to select its corresponding Relative Encoding representation (the third left sub-figure in Fig.1). The absolute direction of the subsequent move (positive X) is then recalled and searched through the selected representation to obtain the corresponding (left) direction to restore its absolute direction (positive X) before the mutation (Fig.3(c)).

4. EXPERIMENTS

4.1 Performance Metrics

As stated in [31, 32], the objective of multimodal optimization is to strike a balance between convergence and diversity. For convergence, the widely used performance metrics have already covered. The energy of the best conformation found ($H(x)$) indicates the best convergence an algorithm can achieve across several runs, whereas the mean

Algorithm 1 Proposed Mutation Method

```
genome: A Relative Encoding genome
random: A random real number from [0,1]
threshold: A real constant from [0,1]
procedure NewMutation(genome)
    savedGenome ← copy of genome;
    if random > threshold then
        AbsolutelyMutate(genome);
    else
        RelativelyMutate(genome);
    end if
    if genome is infeasible then
        genome = savedGenome;
    end if
end procedure

procedure AbsolutelyMutate(genome)
    i ← a random integer from [1, genome.length];
    Randomly change the move i in genome;
    Accordingly change the move i + 1 in genome to
    restore its absolute direction;
end procedure

procedure RelativelyMutate(genome)
    for i from 1 to genome.length do
        if random <= mutation_probability then
            Randomly change the move i in genome ;
        end if
    end for
end procedure
```

and standard deviation of the minimal energy across several runs (*mean* ± *σ*) can report the stochastic convergence behavior of an algorithm. For diversity, however, none of the above performance metrics can reflect. Hence we propose a new performance metric for diversity.

To measure the diversity of the solutions, it is intuitive to count the number of different conformations. Thinking about this measurement deeply, it assumes that different conformations belong to different types of solutions even if they differ by only one residue position. Nevertheless, in our problem, the emphasis is on the formation of non-local H-H bonds (hydrophobic-hydrophobic pairs not adjacent in sequence, but adjacent in lattice). With slight perturbations in the bonds other than H-H bonds, a conformation can spawn a lot of different conformations with the same set of non-local H-H bonds. Similar observation was also arrived by Lopes [24]. Thus we propose counting the number of conformations with different sets of non-local H-H bonds (N). The mean and standard deviation of N across several runs (*mean* ± *σ* of N) are reported in the following experiments.

4.2 Parameter Settings

The parameter settings for the implemented algorithms in all benchmarks are tabulated in Table 1. Crowding (CGA [17]), fitness sharing (SharingGA [13]), and species-conserving (SCGA [23]) techniques have been implemented in the EC4 framework [18]. In particular, the proposed mutation method is equipped in CGA which is then donated as 'CGA-mixed' in this paper. The unified evolutionary algorithm (UN) [18] was also run as a control experiment. For all algorithms other than the crowding algorithms, truncation was applied in survival selection for fairness. Except CGA-mixed, all algorithms adopted the bit flip mutation [18]. With the overlapping generation type and high selection pressure imposed in the survival selection, the mutation probability was set to a high value for achieving global search capability. Thus 0.8 was adopted. Crowding factor was set to population size to avoid replacement error. To be comparable to the state-of-the-art algorithms [27, 9, 7], all algorithms were run 50 times up to 10^5 and 5×10^6 energy evaluations respectively. The benchmarks were taken from [27, 9, 7].

Table 1: Parameter Settings

Parameter	Setting
Population Initialization	Straight line (FFFF....FF)
Population Size	100
Generation Type	Overlapping
Parent Selection	Uniform Deterministic
Survival Selection	Truncation/Crowding
Mutation Type	Bit Flip
Mutation Probability	0.8
Crossover Type	Two Point Crossover
Crossover Probability	1
Random Seed	123
Implementation	EC4 framework [18]

4.3 Results

Table 2 and Table 3 show the experimental results for the multimodal optimization techniques, which were run 50 times up to 10^5 and 5×10^6 energy evaluations respectively. For each benchmark, the performance metrics discussed have been calculated. For instance, looking at Table 2 and sequence s1, CGA-mixed has ever achieved -11 as its minimal energy across 50 runs. On average, CGA-mixed has also achieved -10.8 as its minimal energy and found 97.04 confirmations with different sets of non-local H-H bonds for a run. UN is a simple evolutionary algorithm [18]. It is canonical enough to be a control algorithm without any multimodal optimization techniques. Comparing its results with the other algorithms, it can demonstrate that all multimodal optimization techniques are beneficial to the problem in terms of the performance metrics used. In particular, the crowding techniques with and without the proposed mutation (CGA-mixed and CGA) outperformed the other algorithms. Thus we further compared their results with the results of the state-of-the-art algorithms [27] as shown in Table 4 and Table 5.

Surprisingly, although CGA-mixed and CGA are two relatively simple algorithms, they could still show comparable results with the state-of-the-art algorithms when the termination condition was set to 10^5 energy evaluations. The experiments were further extended to 5×10^6 energy evaluations. CGA-mixed and CGA even showed their competitive edges. Their effectiveness may be largely due to their individual replacement technique: crowding. With this technique, a conformation cannot replace a dissimilar conformation. It gives freedom for all niches to evolve to their respective optima. Diversity is adaptively preserved. In particular, such diversity prevent a population from genetic drift. Useful sub-conformations (like secondary structures [4]) can be preserved, providing the algorithm a long-term sustainability for finding multiple optima at the same time in a single run.

Table 2: Experimental Results of Multimodal Optimization Techniques (10^5 energy evaluations)

Benchmark	Performance	CGA-mixed	CGA	SharingGA [13]	SCGA [23]	UN [18]
s1	H(x)	**-11**	**-11**	**-11**	**-11**	-10
	mean±σ	-10.80±0.40	**-10.88±0.33**	-10.64±0.48	-8.42±1.21	-9.16±0.71
	mean±σ of N	**97.04±1.50**	96.90±1.73	22.44±4.95	51.12±4.47	2.00±1.47
s2	H(x)	**-13**	**-13**	**-13**	-12	-11
	mean±σ	-12.12±0.87	**-12.16±0.91**	-11.36±0.66	-8.52±1.43	-9.66±0.98
	mean±σ of N	**91.08±3.39**	90.72±3.96	18.98±5.18	48.76±4.33	1.80±1.11
s3	H(x)	**-9**	**-9**	**-9**	**-9**	**-9**
	mean±σ	**-9.00±0.00**	**-9.00±0.00**	-8.66±0.56	-7.72±0.93	-7.38±0.81
	mean±σ of N	66.54±5.85	**69.24±5.43**	14.42±4.76	36.86±3.48	1.88±1.12
s4	H(x)	**-18**	**-18**	-17	-15	-15
	mean±σ	**-16.76±0.94**	-16.66±0.96	-14.48±1.22	-10.86±2.22	-12.14±1.23
	mean±σ of N	93.04±3.71	**94.40±4.18**	17.30±5.24	62.78±5.35	1.68±1.13
s5	H(x)	**-29**	**-29**	-25	-17	-22
	mean±σ	**-26.16±1.30**	-25.82±1.22	-21.34±1.67	-10.84±2.86	-17.26±1.84
	mean±σ of N	**97.32±2.00**	97.04±3.42	22.64±6.71	77.38±4.96	1.76±1.02
s6	H(x)	**-28**	**-28**	-24	-15	-21
	mean±σ	**-24.58±1.33**	-24.46±1.31	-19.58±1.91	-10.02±2.62	-16.42±1.81
	mean±σ of N	**97.42±2.19**	96.58±2.70	22.84±6.04	72.94±5.55	1.82±1.21
s7	H(x)	-45	**-48**	-40	-32	-37
	mean±σ	-40.88±2.02	**-41.20±2.18**	-34.82±2.14	-16.26±4.44	-28.60±3.49
	mean±σ of N	99.26±0.69	**99.32±0.77**	26.66±6.46	89.86±3.44	1.38±0.70
s8	H(x)	**-49**	-47	-39	-30	-32
	mean±σ	**-42.62±2.33**	-41.62±2.38	-33.32±2.50	-15.52±3.84	-27.16±2.64
	mean±σ of N	**99.04±0.64**	98.96±0.97	24.90±6.12	87.54±4.17	1.64±0.85

Table 3: Experimental Results of Multimodal Optimization Techniques (5×10^6 energy evaluations)

Benchmark	Performance	CGA-mixed	CGA	SharingGA [13]	SCGA [23]	UN [18]
s1	H(x)	**-11**	**-11**	**-11**	**-11**	**-11**
	mean±σ	**-11.00±0.00**	**-11.00±0.00**	-10.68±0.47	-9.48±1.33	-9.98±0.55
	mean±σ of N	82.60±4.84	**85.44±4.15**	11.28±3.96	53.16±4.29	1.52±0.79
s2	H(x)	**-13**	**-13**	**-13**	**-13**	-12
	mean±σ	**-13.00±0.00**	**-13.00±0.00**	-11.52±0.86	-9.54±1.39	-9.78±0.79
	mean±σ of N	77.04±6.18	**78.70±6.91**	12.14±3.58	50.22±4.74	1.80±0.93
s3	H(x)	**-9**	**-9**	**-9**	**-9**	**-9**
	mean±σ	**-9.00±0.00**	**-9.00±0.00**	-8.74±0.44	-8.08±0.78	-7.72±0.73
	mean±σ of N	33.98±4.81	**42.20±6.40**	8.78±4.10	36.26±3.34	1.64±1.05
s4	H(x)	**-18**	**-18**	-17	-15	-14
	mean±σ	**-18.00±0.00**	**-18.00±0.00**	-14.98±1.10	-11.52±2.62	-12.14±0.99
	mean±σ of N	84.26±6.43	**86.26±5.32**	12.08±4.09	63.6±4.16	1.42±0.64
s5	H(x)	-30	**-31**	-26	-22	-21
	mean±σ	**-28.98±0.55**	-28.70±0.89	-21.86±1.85	-11.88±3.44	-17.46±1.62
	mean±σ of N	93.02±4.58	**93.70±3.92**	16.90±4.73	77.4±4.48	1.50±0.95
s6	H(x)	**-31**	-30	-24	-18	-21
	mean±σ	**-27.78±1.04**	-26.96±1.12	-20.28±1.75	-10.76±3.13	-16.36±1.75
	mean±σ of N	94.26±2.93	**95.80±2.56**	14.20±4.22	74.62±5.34	1.70±0.97
s7	H(x)	**-50**	**-50**	-41	-36	-35
	mean±σ	**-47.42±1.18**	-46.76±1.30	-35.30±2.79	-16.84±5.23	-29.10±2.55
	mean±σ of N	97.92±1.26	**98.02±1.82**	18.92±4.70	90.62±3.27	1.46±0.65
s8	H(x)	**-55**	-52	-41	-31	-32
	mean±σ	**-49.26±1.83**	-47.64±1.88	-34.28±2.42	-16.94±5.01	-27.50±2.53
	mean±σ of N	98.14±1.85	**98.32±1.06**	16.54±4.32	87.72±4.05	1.40±0.76

Table 4: Experimental Results of the state-of-the-art algorithms (10^5 energy evaluations)

		Hybrid GA [7]	IA [9]	MK-EDA2 [27]		TreeEDA [27]		MT-EDA4 [27]		CGA-mixed	CGA
s1	H(x)	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
	mean±σ	-9.84±0.86	-10.90±0.32	**-11.00±0.00**	**-11.00±0.00**	**-11.00±0.00**	-10.96±0.04	**-11.00±0.00**	**-11.00±0.00**	-10.80+0.40	-10.88±0.33
s2	H(x)	-11	**-13**	**-13**	**-13**	**-13**	**-13**	**-13**	**-13**	**-13**	**-13**
	mean±σ	-10.00±0.87	-12.22±0.65	-12.94±0.09	**-13.00±0.00**	-12.86±0.16	-12.96±0.04	-12.70±0.50	**-13.00±0.00**	-12.12±0.87	-12.16±0.91
s3	H(x)	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**
	mean±σ	-8.64±0.69	-8.88±0.48	-8.94±0.06	-8.96±0.04	-8.90±0.09	-8.98±0.02	-8.98±0.02	-8.98±0.02	**-9.00±0.00**	**-9.00±0.00**
s4	H(x)	**-18**	**-18**	**-18**	**-18**	**-18**	-17	**-18**	**-18**	**-18**	**-18**
	mean±σ	-13.72±1.41	-16.08±1.02	-15.66±1.54	-15.48±0.83	-16.34±0.51	-15.00±0.86	-16.32±0.10	-15.02±0.88	**-16.76±0.94**	-16.66±0.96
s5	H(x)	-28	-28	-22	-24	-27	-24	-23	-24	**-29**	**-29**
	mean±σ	-18.90±2.08	-24.82±0.71	-19.66±1.37	-20.52±1.15	-23.62±1.83	-20.68±1.65	-18.44±1.60	-20.22±2.30	**-26.16±1.30**	-25.82±1.22
s6	H(x)	-22	-23	**-30**	-26	**-30**	-26	-28	-24	-28	-28
	mean±σ	-19.06±1.46	-22.08±1.43	-26.30±2.26	-23.38±1.30	-26.00±2.82	-22.08±2.48	**-26.70±1.97**	-22.54±1.27	-24.58±1.33	-24.46±1.31
s7	H(x)	-38	-41	-37	-38	-37	-38	-35	-38	-45	**-48**
	mean±σ	-32.28±3.09	-39.02±0.50	-32.66±3.13	-33.84±2.91	-32.94±1.53	-33.10±3.11	-31.72±2.98	-32.46±3.03	-40.88±2.02	**-41.20±2.18**
s8	H(x)	-36	-42	-42	-40	-44	-34	-37	-37	**-49**	-47
	mean±σ	-30.84±2.55	-39.07±1.20	-36.66±4.02	-34.66±2.60	-34.70±6.87	-30.82±2.97	-32.24±2.47	-30.96±2.47	**-42.62±2.33**	-41.62±2.38

Table 5: Experimental Results of the state-of-the-art algorithms (5×10^6 energy evaluations)

		Hybrid GA [7]	MK-EDA2 [27]	TreeEDA [27]	MT-EDA4 [27]	CGA-mixed	CGA
s1	H(x)	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
	mean±σ	-10.52±0.54	-10.82±0.38	-10.68±0.51	-10.84±0.37	**-11.00+0.00**	**-11.00±0.00**
s2	H(x)	**-13**	**-13**	**-13**	**-13**	**-13**	**-13**
	mean±σ	-11.28±0.90	-12.02±0.94	-11.30±0.85	-11.88±0.93	**-13.00±0.00**	**-13.00±0.00**
s3	H(x)	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**
	mean±σ	-8.54±0.64	-8.96±0.19	-8.92±0.27	**-9.00±0.00**	**-9.00±0.00**	**-9.00±0.00**
s4	H(x)	**-18**	**-18**	**-18**	**-18**	**-18**	**-18**
	mean±σ	-15.76±1.05	-16.40±0.80	-16.24±0.83	-16.50±0.96	**-18.00±0.00**	**-18.00±0.00**
s5	H(x)	-28	-29	-29	-29	-30	**-31**
	mean±σ	-24.60±1.57	-27.24±0.92	-26.88±0.93	-27.06±1.08	**-28.98±0.55**	-28.70±0.89
s6	H(x)	-26	-29	**-31**	-28	**-31**	-30
	mean±σ	-23.02±1.48	-25.70±1.26	-25.94±1.58	-25.74±1.22	**-27.78±1.04**	-26.96±1.12
s7	H(x)	-49	-49	-49	-48	**-50**	**-50**
	mean±σ	-41.18±2.75	-46.30±2.04	-43.78±3.10	-42.00±6.76	**-47.42±1.18**	-46.76±1.30
s8	H(x)	-46	-52	-49	-50	**-55**	-52
	mean±σ	-40.40±2.50	-46.78±2.28	-43.72±2.43	-45.64±2.03	**-49.26±1.83**	-47.64±1.88

4.4 Effect of Crowding Factor

To demonstrate the effectiveness of crowding, CGA and CGA-mixed were run 50 times up to 10^5 energy evaluations with different values of crowding factor. The results are depicted in Fig.4. The horizontal axis denotes different values of crowding factor used, whereas the vertical axis denotes the corresponding means of N. The results showed that the mean of N is directly proportional to the value of crowding factor, indicating that more diverse conformations can survive if the value of crowding factor gets larger. Less replacement errors occur when crowding factor gets larger [25].

5. CONCLUSION

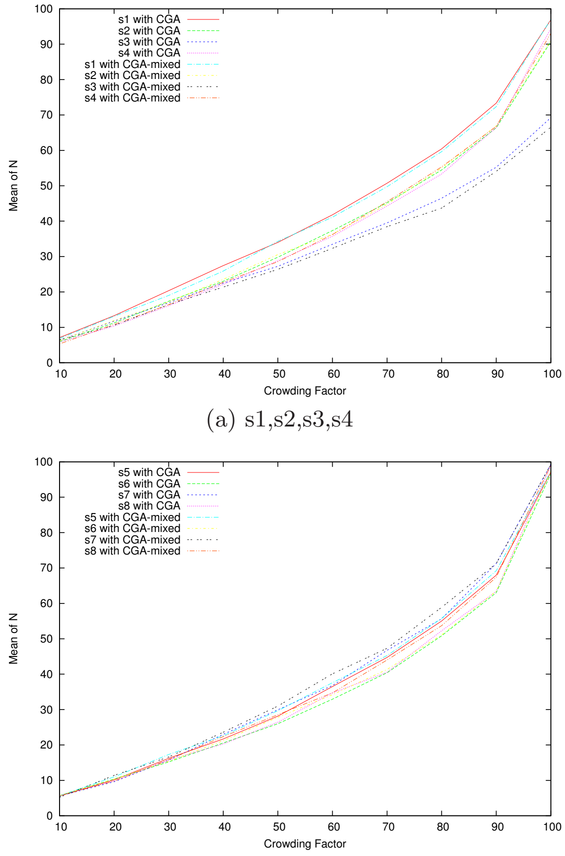
In this paper, we have modeled the protein structure prediction problem as a multimodal optimization problem. To foster its development, several mutlimodal optimization techniques have been implemented and tested. In particular, we have proposed a new mutation approach mixing AM and RM together as a pair of complementary operations. A new performance measure (mean±σ of N) has also been proposed for the multimodal optimization problem. The experimental results indicated that the crowding technique performed the best among the other multimodal optimization techniques tested. It even performed better than the other state-of-the-art algorithms, although it is just a simple technique. Such surprising results may also provide some biological implications for scientists. For instance, the importance of the existences of intermediate sub-conformations could be examined in the pathways provided by the multimodal optimization techniques.

In the future, AM and RM could be adaptively selected like mutlimeme algorithm [19], instead of a single threshold. More multimodal optimization techniques could also be examined.

The source code is released at http://pc89075.cse.cuhk.edu.hk:8080/myapp/GECCO2010-PSP-LatticeModels.zip

6. ACKNOWLEDGMENTS

The authors are grateful to anonymous reviewers for their valuable comments. The authors would also like to thank Benjamin Ching-Man Tse for his contribution. This research is partially supported by the grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. 414107 and 414708).



(a) s1,s2,s3,s4

(b) s5,s6,s7,s8

Figure 4: Sensitivity Analysis of Crowding Factor

7. REFERENCES

[1] S. Aluru. *Handbook of Computational Molecular Biology (Chapman & All/Crc Computer and Information Science Series)*. Chapman & Hall/CRC, 2005.

[2] C. B. Anfinsen. Principles that Govern the Folding of Protein Chains. *Science*, 181(4096):223–230, 1973.

[3] D. Baker. A surprising simplicity to protein folding. *Nature*, 405(6782):39–42, May 2000.

[4] D. Baker and A. Sali. Protein Structure Prediction

and Structural Genomics. *Science*, 294(5540):93–96, 2001.

[5] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (hp) is np-complete. In *RECOMB '98: Proceedings of the second annual international conference on Computational molecular biology*, pages 30–39, New York, NY, USA, 1998. ACM.

[6] R. Bitello and H. S. Lopes. A differential evolution approach for protein folding. In *Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB '06. 2006 IEEE Symposium on*, pages 1–5, Toronto, Ont.„ Sept. 2006.

[7] C. Cotta. Protein structure prediction using evolutionary algorithms hybridized with backtracking. In *IWANN '03: Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks*, pages 321–328, Berlin, Heidelberg, 2003. Springer-Verlag.

[8] G. A. Cox, T. V. Mortimer-Jones, R. P. Taylor, and R. L. Johnston. Development and optimisation of a novel genetic algorithm for studying model protein folding. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling*, 112(3):163–178, 2004.

[9] V. Cutello, G. Nicosia, M. Pavone, and J. Timmis. An immune algorithm for protein structure prediction on lattice models. *IEEE Transactions on Evolutionary Computation*, 11(1):101–117, Feb. 2007.

[10] K. A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, March 1985.

[11] Y. Duan and P. A. Kollman. Computational protein folding: from lattice to all-atom. *IBM Syst. J.*, 40(2):297–309, 2001.

[12] S. D. Flores and J. Smith. Study of fitness landscapes for the HP model of protein structure prediction. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 4, pages 2338–2345, Dec. 2003.

[13] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic algorithms and their application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[14] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, Sep 1993.

[15] T. Hoque, M. Chetty, and L. S. Dooley. A guided genetic algorithm for protein folding prediction using 3d hydrophobic-hydrophilic model. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2339–2346, Vancouver, BC„ 2006.

[16] T. R. Jahn and S. E. Radford. Folding versus aggregation: polypeptide conformations on competing pathways. *Arch. Biochem. Biophys.*, 469:100–117, Jan 2008.

[17] K. A. D. Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975.

[18] K. A. D. Jong. *Evolutionary Computation. A Unified Approach.* MIT Press, Cambridge, MA, USA, 2006.

[19] N. Krasnogor, B. Blackburnem, J. Hirst, and E. Burke. Multimeme algorithms for protein structure prediction. In *7th International Conference Parallel Problem Solving from Nature*, volume 2439 of *Springer Lecture Notes in Computer Science*, pages 769–778, Granada, Spain, September 2002. PPSN, Springer Berlin / Heidelberg. ISBN 3-540-44139-5.

[20] N. Krasnogor, W. Hart, J. Smith, and D. Pelta. Protein structure prediction with evolutionary algorithms. In *International Genetic and Evolutionary Computation Conference (GECCO99)*, pages 1569–1601. Morgan Kaufmann, 1999.

[21] C. Levinthal. Are there pathways for protein folding? *J. Chem. Phys.*, 65:44–45, 1968.

[22] H. Li, R. Helling, C. Tang, and N. Wingreen. Emergence of Preferred Structures in a Simple Model of Protein Folding. *Science*, 273(5275):666–669, 1996.

[23] J. P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evol. Comput.*, 10(3):207–234, 2002.

[24] H. S. Lopes. Evolutionary algorithms for the protein folding problem: A review and current trends. *Computational Intelligence in Biomedicine and Bioinformatics*, pages 297–315, 2008.

[25] S. W. Mahfoud. Simple analytical models of genetic algorithms for multimodal function optimization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, page 643, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[26] A. L. Patton, W. F. Punch, III, and E. D. Goodman. A standard ga approach to native protein conformation prediction. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 574–581, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[27] R. Santana, P. Larranaga, and J. A. Lozano. Protein folding in simplified models with estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 12(4):418–438, Aug. 2008.

[28] A. Shmygelska and H. Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6(1):30, 2005.

[29] R. Unger and J. Moult. Genetic algorithm for 3d protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[30] R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *J. Mol. Biol.*, 231:75–81, May 1993.

[31] K.-C. Wong, K.-S. Leung, and M.-H. Wong. An evolutionary algorithm with species-specific explosion for multimodal optimization. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 923–930, New York, NY, USA, 2009. ACM.

[32] K.-C. Wong, K.-S. Leung, and M.-H. Wong. Effect of spatial locality on an evolutionary algorithm for multimodal optimization. In *EvoApplications 2010, Part I, LNCS 6024*. Springer-Verlag, 2010.