Multi-objective evolutionary strategy approaches for protein structure prediction

by

Shuangbao Song

A dissertation
submitted to the Graduate School of Science and Engineering for Education
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Engineering

University of Toyama
Gofuku 3190, Toyama-shi, Toyama 930-8555 Japan

2019
(Submitted December 16, 2019)

Acknowledgements

I would like to deeply thank the various people who, during my study and research, gave me with useful and helpful assistance. Without their care and consideration, this thesis would likely not have finished.

Firstly, I would like to express my sincere gratitude to my advisor Professor Zheng Tang, who introduced me to the significant and fascinating world of computational intelligence, for his support and continuous encouragement. Without his help and encouragement, I would never have completed this degree. I would also like to express my sincere gratitude to Associate Professor Shangce Gao, who introduced me to the research field of protein structure prediction, for his great support and sharing with me his extensive experience.

Besides my advisor, I would like to thank all of the members of the Intelligent Information Systems Research Lab in University of Toyama, for all their help and friendship that made this time much more enjoyable.

Last but not the least, I would like to thank all of my family, for their unconditional love, support, and encouragement through this process, through all my study process.

Abstract

The problem of predicting the three-dimensional structure of a protein from its one-dimensional sequence has been called the “holy grail of molecular biology”, and it has become an important part of structural genomics projects. Despite half-century’s unremitting efforts, the prediction of protein structure from its amino acid sequence remains a grand challenge in computational biology and bioinformatics. Two key factors are crucial to solving the protein structure prediction (PSP) problem: an effective energy function and an efficient conformation search strategy.

In my research of defending PhD, I focus on modeling the PSP problem as a multi-objective optimization problem, and use an evolutionary strategy to solve the problem. A method MO3 and its improved version AIMOES, were proposed during my research of defending PhD. They are illustrated as follows:

(1) Firstly, in MO3, we propose a multi-objective evolutionary algorithm. We decompose the protein energy function Chemistry at HARvard Macromolecular Mechanics force fields into bond and non-bond energies as the first and second objectives. Considering the effect of solvent, we innovatively adopt a solvent-accessible surface area as the third objective. We use 66 benchmark proteins to verify the proposed method and obtain better or competitive results in comparison with the existing methods. The results suggest the necessity to incorporate the effect of solvent into a multi-objective evolutionary algorithm to improve protein structure prediction in terms of accuracy and efficiency.

(2) Secondly, in AIMOES, we model the PSP as a multi-objective optimization problem. A three-objective evolution algorithm called AIMOES is proposed. AIMOES adopts three physical energy terms: bond energy, non-bond energy, and

solvent accessible surface area. In AIMOES, an evolution scheme which flexibly reuse past search experiences is incorporated to enhance the efficiency of conformation search. A decision maker based on the hierarchical clustering is carried out to select representative solutions. A set of benchmark proteins with $30 \sim 91$ residues is tested to verify the performance of the proposed method. Experimental results show the effectiveness of AIMOES in terms of the root mean square deviation (RMSD) metric, the distribution diversity of the obtained Pareto front and the success rate of mutation operators. The superiority of AIMOES is demonstrated by the performance comparison with other five state-of-the-art PSP methods.

This thesis is organized as follows: Chapter 1 gives a brief introduction about the PSP problem and multi-objective optimization. Chapter 2 presents some important concepts. Chapter 3 presents the energy function used in these two methods. In Chapter 4, we shows the multi-objective evolutionary strategy where solvent effect are incorporated into, i.e. MO3, for solving the PSP problem. The experimental results of MO3 are also shown in this chapter. Then, in Chapter 5, the archive information assisted multi-objective evolutionary strategy, i.e. AIMOES, for solving the PSP problem is described. Finally, we draw the conclusions of this thesis in Chapter 6.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Proteins are large biomolecules and the building blocks of life. They are fundamental elements of cells and perform many biological functions, such as transporting biomolecules, providing structural support and aiding the biochemical reaction. The basic structure of a protein is a linear chain of 20 difference types of amino acids, held together by amide bond. Different amino acid sequences make up different proteins uniquely and determine the three-dimensional structure of the protein uniquely. As the three-dimensional structure of a protein controls its basic activity and function, determining the structure of a protein is important in the biological research [1].

Two common ways to obtain the three-dimensional structure of proteins are X-rays and NMR spectroscopy. However they are all time-consuming and expensive. Alternatively, a promising way is to use computational approach to predict the three-dimensional structure of a protein from its sequence [2, 3, 4]. In the early 1970s, Nobel Prize winner Anfinsen suggested that it is sufficient to predict the three-dimensional structure of a protein staring from its amino acid sequence [5]. This problem is thus called protein structure prediction (PSP) problem. Two main motivations drive the researches of PSP:

(1) The digitization gap between sequence entries and structure entries is huge. Till November 2016, the UniProtKB protein sequence database contains over 60 million sequence entries [6]. However, there are only about 0.2% structure entries released in the Protein Data Bank (PDB) [7].

(2) Knowledge of the three-dimensional structure of the protein can help in un-

derstanding its function and role in life. It is thus extremely important to reveal the atomic-level structural information of all found protein sequence entries. Moreover, the disposal of the PSP problem can also enable us to understand the principles of proteins folding in nature, thus improving other biotechnologies [8].

Protein structure prediction methods can be divided roughly into two categories: template-based modeling (TBM) and free modeling (FM) [9]. If a similar protein of target protein is identified from the PDB, the TBM is likely to be used, where the three-dimensional structure of the target protein is reconstructed according to its similar protein (the "template"). For example, SWISS-MODEL is a widely-used and accurate homology modeling method to generate reliable protein structures [10]. I-TASSER is an on-line platform for protein structure and function prediction based threading [11]. On the other hand, if no template protein is available, the FM should be used [12]. Among FMs, fragment assembly based methods are the most successful, such as Rosetta [13] and QUARK [14]. They optimize a protein conformation by assembling small fragments extracted from native protein structures into some selected insertion windows of it. Nevertheless, fragment assembly based methods are sometimes deemed to use information from existing PDB structures a bit too much. Although FM are generally more imprecise than TBM, it is considered to help us reveal the principles of proteins folding more directly, and thus has greater values in theory [15, 16, 17, 18].

A typical FM (also called ab initio prediction) is made up of a conformation space search strategy and a designed energy guiding search process. It follows the thermodynamic hypothesis: a native protein structure stays stably in the state of the minimum free energy in a suitable environment [5]. However, the performance of existing FMs are limited due to the inaccuracy of the (physical or statistic) energy function and the huge size of the conformation search space [1]. In addition, the landscape of the designed energy function is very rugged and complex due to its multimodal nature with many local minimum. FM must struggle with these difficulties for high prediction accuracy. As David and Andrej pointed out in [19], the key factors of high performing FM are the design of an accurate energy function and the implementation of an efficient conformation search algorithm.

Convenient FMs treat PSP as a single-objective optimization problem (SOOP) [20]. They usually incorporate different energy terms as a linear combination with different weights. However, the values of those weight parameters in the energy function is very difficult to be optimized, thus limiting the efficiency of these methods. An alternative way is modeling PSP as a multi-objective optimization problem. It is empirically viewed that the transformation of PSP into a multi-objective optimization problem (MOOP) may increase the problem's difficulty. Actually, by incorporating efficient search mechanisms and selections of non-dominated solutions, the treatment of PSP as a MOOP can generate more fruitful results. In recent years, many effective multi-objective evolution algorithms (MOEA) [21, 22, 23, 24] have been proposed to perform PSP. Cutello et al. proposed a two-objective evolutionary algorithm to perform PSP by optimizing two physical energy functions which are in conflict with each other [25]. Brasil et al. proposed a MOEA, called MEAMT, to solve PSP by dealing with four objective functions through the combination thereof [26]. These works have showed the advantages of modeling PSP as a MOOP.

Evolutionary algorithm (EA) is inspired by biological evolution with a stochastic search for a given problem. It is designed as a computer-based algorithm which aims to solve global optimization. In recent years, a great number of EAs have been proposed in the literature, such as particle swarm optimization [27, 28], differential evolution [29], gravitational search algorithms [30, 31, 32], artificial bee colony algorithm [33], cuckoo search [34], and brain storm optimization [35]. Due to their powerful computation performance, these EAs have achieve great success in many practical problems, including protein structure prediction [28, 36, 37, 38], training dendritic neuron models [39, 40, 41], gene selection [42], drug design [43, 44], and so on. These works have showed that EAs are a very powerful and effective technique for solving many complex problems.

Our proposed approach MO3 follows a free modeling approach [1]. We decompose PSP into two main subproblems. The first one is to find a free energy function that can accurately distinguish the native state from similar conformations. The second one is to design an algorithm that explores the conformation space to find the global minimum of this energy. For these two subproblems, we take the following measures to improve the performance of the proposed method, i.e., taking the factor of solvent into consideration to amend the energy function and using a multi-objective evolutionary approach to improve the search capability.

The contribution of MO3 is fourfold: 1) incorporating solvent effect for solving PSP; 2) modeling PSP as a three-objective optimization problem and designing a multi-objective evolutionary algorithm to solve it; 3) testing a set of benchmark proteins to evaluate the proposed approach; 4) inspecting and revealing the conflict between solvent effect and protein energy functions.

To achieve a high-resolution PSP algorithm, it is crucial to execute efficient conformation search since the search landscape of energy functions is so rugged that all algorithms will face the problem of being trapped in local minimum during search process. To alleviate this problem, several methods by combining the convenient optimization (e.g., Monte Carlo simulation) with evolutionary algorithms have been proposed [45, 46, 14]. Despite achieving some successes, these methods involve an iterative reproduction process with a large number of evolutions which makes the procedure highly complicated, computational time-consuming and difficult to be well-tuned [2, 47].

In AIMOES, we propose an archive information assisted MOEA (namely AIMOES) for ab initio PSP. The three-objective energy function suggested in [48] is used to guide the search process. The physical energy function, i.e., Chemistry at Harvard Macromolecular Mechanics (version22) [49], is decomposed into bond and non-bond energy as the first and secondary objective. To reflect the effect of solvent, SASA is used as the third objective. Furthermore, an archive information assisted non-dominated solutions generation scheme is proposed to flexibly reuse past search experiences and thus to enhance the efficiency of conformation search. Regarding the information reuse technology, it is generally realized from two aspects: (1) Reusing the model based information which is generated during solving past problems. For example, Hauschild et al. proposed a model building way in the hierarchical Bayesian optimization algorithm, by reusing probabilistic models obtained from solving other similar problems [50]. Iqbal et al. proposed a learning classifier system reusing useful building blocks extracted from small-scale problems [51]. Then, the accumulated information enables the method to be applied to solve complex large-scale problems effectively. (2) Reusing the past optimized solutions or knowledge extracted from them. For example, Louis and McDonnell presented a genetic algorithm which reuses the optimized solutions of past problems by injecting them into the population periodically [52]. In [53], Feng et al. proposed an evolutionary search paradigm for heterogeneous problems, which can learn structured knowledge from search experience by means of a single layer autoencoder. In this study, we innovatively design a new scheme to reuse the information stored in the archive which is constituted by non-dominated solutions obtained by the algorithm. Then the information (especially the backbone torsion angles) is injected into the current solution by means of a mutation operator to generate a new solution. It can be expected that the new generated solution is more promising. Finally, a near-optimal and well-distributed Pareto front is acquired after implementing the evolutionary search with sufficient iterations and the representative solutions are selected from the Pareto front by a decision maker algorithm. To verify the performance of AIMOES, a set of benchmark proteins taken from PDB is used as the test suit. The experimental results in terms of the root mean square deviation (RMSD) between the predicated protein structure and the native one, the success rate of the proposed information reuse mutation operator and the distribution properties of the obtained Pareto front are analyzed. It is demonstrated that AIMOES can produce better or very competitive results in comparison with other five state-of-the-art evolutionary algorithms.

Chapter 2

Materials

2.1 Multi-objective optimization problem (MOOP)

A multi-objective optimization problem (MOOP) involves optimizing more than one objective simultaneously [54]. It can be described as follows:

$$\begin{aligned}&\mathit{Minimize}\ \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})]^T,\\&\mathit{subject\ to}\ \mathbf{x} \in X,\end{aligned} \tag{2.1}$$

where integer k (≥ 2) is the number of objective functions. X is the decision variable space (a feasible set of decision vectors), and an element $\mathbf{x} \in X$ represents a feasible solution. $\mathbf{f}(\mathbf{x})$ is a vector of objective function values, and $f_i(\mathbf{x})$ is the i-th objective function, $i \in \{1, 2, ..., k\}$.

In contrast to a single-objective optimization problem, the situation of comparing two solutions is more complex in MOOP. First, we should introduce the concept of Pareto dominance. In mathematical terms, a feasible solution $\mathbf{x}_1$ is said to dominate another $\mathbf{x}_2$, denoted by $\mathbf{x}_1 < \mathbf{x}_2$, if

$$\begin{aligned}&1)\ \ \forall i \in \{1, 2, ..., k\},\ \ f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2);\ \ \ \mathit{and}\\&2)\ \ \exists j \in \{1, 2, ..., k\},\ \ f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2).\end{aligned}$$

A solution $\mathbf{x}^* \in X$ is called Pareto optimal if

$$\neg\exists \mathbf{x}' \in X,\ \mathbf{x}' < \mathbf{x}^*. \tag{2.2}$$

The Pareto optimal set P can be defined as the set of all Pareto optimal solutions:

$$P = \{\mathbf{x}^* \in X \mid \neg\exists\ \mathbf{x}' \in X, \mathbf{x}' < \mathbf{x}^*\}. \tag{2.3}$$

For a given MOOP, the image of the Pareto optimal set P is often called the Pareto front Γ. In mathematical terms, it can be defined as

$$\Gamma = \{\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})]^T \mid \mathbf{x} \in P\}. \tag{2.4}$$

However, in a practical problem, it is generally impossible to obtain a true Pareto optimal set via a multi-objective optimization method. Instead, we may obtain its approximation. We are interested in how to generate its best approximation set. The goal of multi-objective optimization is clear: to generate a good approximation set of solutions that is convergent to P and diverse [55].

A considerable number of successful applications of multi-objective optimization have been reported [56, 57, 58]. It has been extensively studied in the well-known problems such as a traveling salesman problem [59], bin packing problem [60] and vehicle routing problem [61]. Also, it has been proposed to deal with bioinformatic problems, such as gene regulatory networks [62] and feature selection problem [63]. Such prior work shows that using multi-objective optimization rather than single-objective optimization to solve complex problems has been a current and promising research direction.

As a final note, the goal of this work is to obtain a good set of protein structure solutions using a multi-objective evolutionary algorithm. More knowledge about the set will help decision makers in selecting the best competitive solution that is closest to the native protein conformation.

2.2 Protein representation

Proteins consist of at least one chain of 20 different amino acids, and are very complex to be described at the molecular scale. They are represented as atomic coordinates for all (or heavy) atoms in PDB format file [7]. In contrast, the representation of torsion angles is widely used to solve the PSP problem [64, 14, 25], due to the fact that it can strongly reduce the degree of freedom of protein conformation, by means of setting the bond lengths and bond angles to ideal values.

We use the all-atom model [49] including all hydrogens to represent the protein structure. Different from a reduced model [65], the all-atom model doesn't exclude some insignificant atom or simplifies side-chain as a center of mass. The energy function used in the work is based on physical terms and the contributions of every atom are all considered, making the representation more accurate. In addition, the all-atom representation can avoid steric clash effectively, because these steric clash can lead to unreasonable values (usually very large) of some physical energy terms of a conformation, making it less competitive.

Protein consists of amino acid sequence, and the conformation of every amino acid can be uniquely determined by torsion angles. The torsion angles considered in the all-atom representation model consist of backbone angles (ϕ, ψ, ω) and side-chain angles (χ_i, $i \in \{1, 2, 3, 4\}$). Fig. 2.1 shows the representation of torsion angles for isoleucine amino acid. In this way, a protein structure can be represented as a one-dimensional vector of these torsion angles.

From the view of a search algorithm, the search space of torsion angles is very large. It is necessary and wise to put restrictions on these torsion angles, which can help the search algorithm converging more quickly than the ones with no restrictions [66]. We use the secondary structure information obtained from the PSIPRED protein structure prediction server [67]. PSIPRED can classify a residue in the protein sequence into three class of secondary structure element: helix (H), sheet (E), coil (C). For backbone torsion angles ϕ and ψ, the constraints are shown in Table 2.1.

As described in [68], torsion angle ω is mainly observed to be close to 180° in

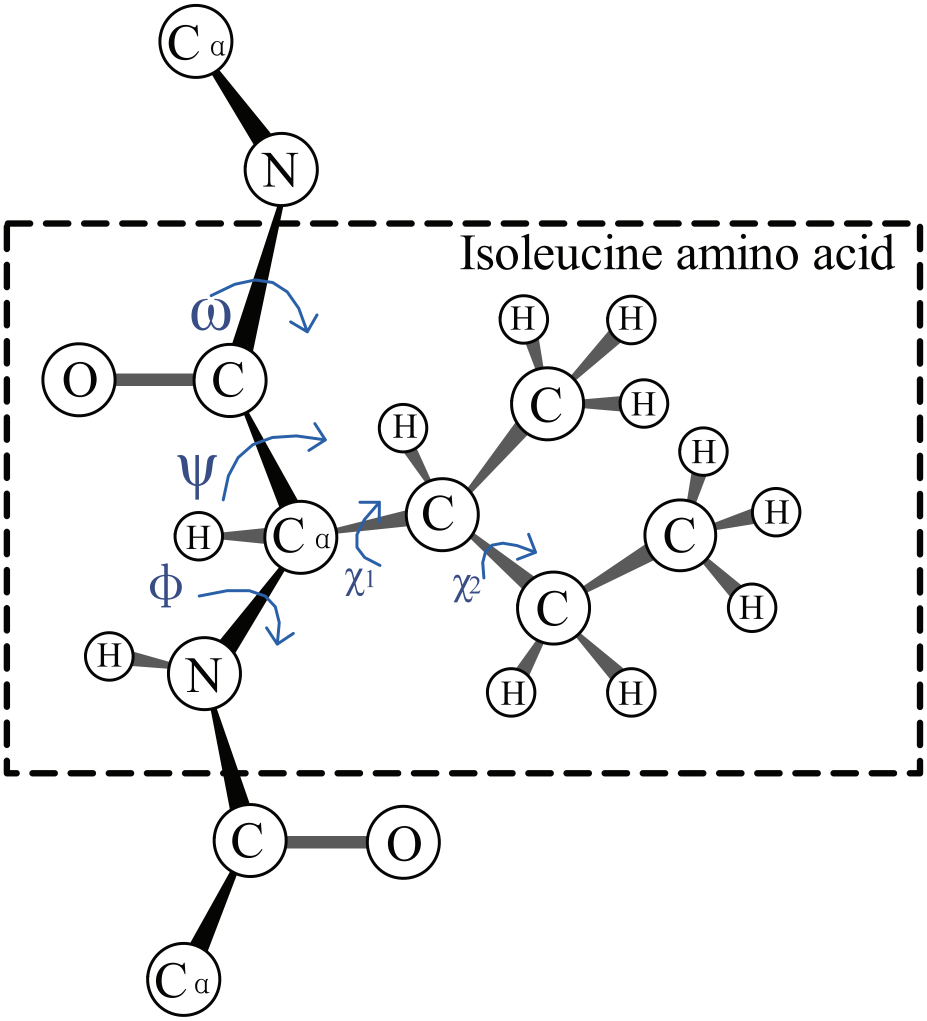


Figure 2.1: Representation of torsion angles for isoleucine amino acid.

protein structures, rather than 0°, thus, all ω are set to 180° in this study. Besides, for side-chain torsion angles χ_i, the restrictions obtained from rotamer library [69] is used. It should be noted that, many restrictions on a dihedral angle space also mean that a significant part of the solution space cannot be sampled, and that the native structure may be unattainable. The torsion angle space sampled by prediction method is largely restricted in one way or another, and this is a limitation of all these methods [25, 70, 71].

Table 2.1: Constraints of secondary structure.

	ϕ	ψ
helix (H)	$[-67°, -47°]$	$[-57°, -37°]$
sheet (E)	$[-130°, -110°]$	$[110°, 130°]$
coil (C)	$[-180°, 180°]$	$[-180°, 180°]$

2.3 Metrics

In this section, we introduce the metrics for evaluating the solution quality and present the results obtained by using the proposed method. We employ two frequently used metrics, root mean square deviation (RMSD) and global distance test-total score, to evaluate the similarity between the predicted conformation and the native structure.

RMSD is calculated as:

$$RMSD_{(a,b)} = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}}, \tag{2.5}$$

where structures a and b have been optimally superimposed by using the Kabsch rotation matrix [72]. d_i is the distance between atom i of structure a and the matched one of b. n is the number of matched atoms. Thus, a smaller RMSD value corresponds to a better structure.

The global distance test (G) score is a measure of similarity between two protein structures and has become a standard evaluation measure in the field of protein prediction. It is computed as follows:

$$G = \frac{100(C_1 + C_2 + C_3 + C_4)}{4M}, \tag{2.6}$$

where C_1, C_2, C_3 and C_4 are the numbers of aligned residues under distance cutoffs of $\vartheta/4$, $\vartheta/2$, ϑ, and $2\times\vartheta$, respectively. The metrics G_4 is a computed value by setting ϑ to 4Å in G. M is the number of amino acids in the compared proteins. The G score has a value of 0∼100, and a larger G_4 value corresponds to a better structure [73, 74].

Now GDT analysis has been the primary method of evaluation in the Critical Assessment of Structure Prediction (CASP) experiment (`http://www.predictioncenter.`

org/index.cgi). It is a cumulative plot of atom C_α accuracy, calculating the percent of aligned residues under distance cutoffs of 0.5Å, 1.0Å, 1.5Å, ... , 10.0Å.

2.4 General procedures of FM

PSP is very difficult to solve because the size of the conformational space to be searched is vast [75] and because the accurate calculation of the free energies of protein conformations in solvent is difficult [76]. Moreover, the search problem in PSP has been demonstrated to be NP-complete [77, 78], i.e., no polynomial time solution algorithm is possible. To make the paper self-explanatory, we first introduce some basic concept. The problem can be solved with four main procedures:

1. Representing the three-dimensional conformation of a protein;
2. Selecting a fitness function to evaluate the conformation; and
3. Designing a search strategy to obtain a desired solution, guided by the fitness function.
4. Performing a decision-making scheme to select solutions from the obtained Pareto optimal set.

Chapter 3

Multi-objective energy function

The conventional FMs treat the PSP as a SOOP, which usually uses a single-objective energy function to filter better structure during search procedure [64, 14]. These energy functions can be divided into two types: physics-based energy functions (based on physical measurements) and knowledge-based energy functions (based on statistic information from PDB library) [79]. Unfortunately, energy functions are not accurate enough to distinguish the native structure ideally. Thus, unremitting efforts have been put into designing more accurate energy functions. The physical ones (e.g., CHARMM [80], OPLS [81]) and statistical ones (e.g., DFIRE [82], RW [83]) are all considered to be state-of-the-art. In the early twenty-first century, Baker et al. [19] has pointed out that designing an effective energy function is one of the important keys to solve the PSP problem .

As argued in [47], a well-designed energy function usually leads to more difficulties because of more rugged landscape. The landscape is very complex with many local minimums. A search process is easily trapped in these local minimum states. A way to reduce local optima is transforming a SOOP into a MOOP [84]. Many works have shown that multi-objective optimization methods are more powerful than single-optimization methods to solve complex problems in the field of computational biology and bioinformatics [85, 86, 87, 62].

Two ways are usually used to transform a SOOP into a MOOP: (1) decomposing the original objective function into several items and (2) adding new objective functions correlated with the original objective function. In normal cases, those objective functions in MOOP should be in conflict with each other [88] and it is a precondition of applying MOEAs. In this work, we use the decomposition and addition of the current energy function to design a new one with three objectives. Firstly, the concept of MOOP and three-objective energy function are described.

3.1 Decomposing CHARMM22 energy function

As is well known, existing methods for predicting protein structures are based on Anfinsen's dogma, also known as the thermodynamic hypothesis [5]. They assume that the native state of a protein is the state of the lowest free energy in a given environment. For free modeling, it is necessary and important to select proper energy functions to evaluate the predicted conformation of a protein.

Many types of energy functions exist and can be divided into two types: statistical effective energy functions (SEEFs) [66] and physical ones (PEEFs) [89]. The former are based on the statistical observations of known protein structures. The latter are an approximation of the true energy functions of proteins, such as Chemistry at HARvard Macromolecular Mechanics (CHARMM) and AMBER [90]. In contrast to SEEFs, the latter consist of molecular mechanics energy functions, which are true physical measurements. PEEFs are supposedly fit for ab initio calculations. We use a type of PEEF, CHARMM force fields (version 22), to evaluate the conformation of a protein.

CHARMM is a widely used set of force fields for molecular dynamics, and it was first described by Brooks et al. in [91]. As a version of CHARMM, CHARMM22 is commonly used as a protein force field [49]. The general form of the potential energy function in CHARMM22 is given as follows:

$$
\begin{aligned}
E \;=\; & \sum_{stretches} k_b(b-b_0)^2 \;+\; \sum_{angles} k_\theta(\theta-\theta_0)^2 \;+ \\
& \sum_{dihedrals} k_\phi[1+cos(n\phi-\delta)] \;+ \\
& \sum_{improper} k_\omega(\omega-\omega_0)^2 \;+\; \sum_{Urey-Bradley} k_u(u-u_0)^2 \;+ \\
& \sum_{Van-der-Waals} \varepsilon_{ij}\left[(\frac{R_{ij}}{r_{ij}})^{12} - 2(\frac{R_{ij}}{r_{ij}})^6\right] \;+ \\
& \sum_{electrostatic} \frac{q_i q_j}{e r_{ij}}.
\end{aligned}
\tag{3.1}
$$

It consists of seven terms:

1. bond stretches, where k_b is the bond force constant, b is the bond length, and b_0 is the equilibrium bond length.

2. bond angles, where k_θ is the angle force constant, θ is the valence angle among 3 bonded atoms, and θ_0 is the equilibrium angle.

3. dihedrals (also called torsion angles), where k_ϕ is the dihedral force constant, n is the multiplicity of the function, ϕ is the torsion angle, and δ is the phase shift.

4. improper angles, where k_ω is the force constant, ω is the improper angle, and ω_0 is the equilibrium improper angle.

5. Urey-Bradley component, where k_u is the respective force constant, u is the distance between atoms 1 and 3 (two atoms separated by two covalent bonds), and u_0 is the equilibrium distance.

6. van der Waals potential energy, which is calculated via the Lennard-Jones potential (also referred to as the 12-6 potential). ε_{ij} is the depth of the potential well, r_{ij} is the distance between a pair of atoms i and j, and R_{ij} is the distance at which the potential reaches its minimum.

7. electrostatic energy, where q_i and q_j are the point charges, e is the dielectric constant, and r_{ij} is the distance between a pair of atoms i and j.

Note that, in the sixth term, the values of ε_{ij} and R_{ij} are obtained according to individual atom types. In current CHARMM force fields, ε_{ij} and R_{ij} for the interacting atoms are obtained by referring to combination rules, i.e.,

$$\varepsilon_{ij} = \sqrt{\varepsilon_{ii}\varepsilon_{jj}}, \tag{3.2}$$

$$R_{ij} = \frac{R_i + R_j}{2}. \tag{3.3}$$

The potential energy between two atoms arises from a balance between repulsive and attractive forces. Specifically, when the distance between two atoms is too small, the energy is quite large. We will show later in this paper that the van der Walls energy of a poor protein structure can be very high.

A single-objective optimization problem is transformed into a multi-objective optimization problem by decomposing the original objective function into multiple ones or by adding new supplementary objectives [84]. To perform PSP as a multi-objective optimization problem, we first decompose the original potential energy function CHARMM22. As suggested by Brooks et. al. [80], the terms in it can be divided into two types: internal and non-bonded terms. The former (also called bond energy) include bond stretches, bond angles, dihedrals, improper angles and Urey-Bradley component. The latter (also called non-bond energy) include van der Waals potential energy and electrostatic energy. Following this view, we divide CHARMM22 into two types: bond and non-bond energy. Moreover, we implement them as the first and second objectives during a multi-objective optimization process. Another reason for decomposing the energy function is that the non-bond energy charge can hide bond energy terms because it has a larger change range, to be shown later. This answers why we should separate the non-bond energy from the bond ones [92, 93, 94, 95].

3.2 Using SASA as the third objective function

A solvent-accessible surface area (SASA) is the surface of a biological molecule that is accessible to a solvent. Lee et al. presented the first algorithm for calculating the SASA of a molecule in 1971 [96]. A typical method is to use a solvent sphere whose radius is typically 1.4Å to probe the surface of a molecule. Therefore, SASA can be viewed as the surface area of a union of balls.

Initially, SASA calculation was applied to study the protein folding problem and hydrophobicity. Because the surface and shape determine how biological molecules interact with other molecules, SASA can partially reflect the role of surfaces in physiological processes. Currently, SASA calculation has contributed to molecular biology studies, including DNA-protein interactions, protein folding, protein secondary structure prediction [97] and protein tertiary structure prediction [4].

All SFFEs and PFFEs can be considered as approximations to the true (unknown) protein potential energy. To make some corrections, we should take other factors into consideration. In [98, 99], the effective energy function refers to the free energy of the system (protein and solvent). In other words, an effective energy function consists of the inter-molecular energy of the protein plus the solvent free energy. For this purpose, there are many extended versions of CHARMM that have been improved by incorporating the influence of the solvent explicitly or implicitly. As a result, we have explicit ones, e.g., TIP3P water model [100] and implicit ones, e.g., Gaussian solvation free energy model [101] and SCPISM continuum model [102].

The earliest and simplest implicit solvent models are SASA models [96]. This type of models respresent solvent as a continuous medium and assume that the solvent free energy of each part of a molecule is proportional to its SASA. Wesson and Eisenberg expressed the solvation free enery term as a sum over individual atomic contributions:

$$G_{sol} = \sum_{atom\ i=1} \delta_i A_i, \tag{3.4}$$

where δ is an atomic solvation parameter depending on the atom type. A_i is the

SASA of the atom [103, 104]. Moreover, in the Generalized Born / Surface Area (GB/SA) model [105], the solvation free energy G_{sol} is calculated by summing three terms: solvent-solvent cavity term G_{cav}, solute-solvent van der Waals term G_{vdW}, and a solute-solvent electrostatics polarization term G_{pol}:

$$G_{sol} = G_{cav} + G_{vdW} + G_{pol}. \tag{3.5}$$

For nonpolar molecules, $G_{pol} = 0$ and G_{sol} can be treated as linear approximation of their SASA:

$$G_{cav} + G_{vdW} = \sum_{atom\ i=1}^{N} \delta_i A_i, \tag{3.6}$$

where δ_i is an empirical atomic solvation parameter, A_i is the SASA for the atom. This work includes the SASA of a protein conformation as the third objective function, which reflects the effect of the solvent implicitly. The smaller SASA, the better conformation.

As suggested in [99, 19, 48], in the molecular system an effective energy function should also take the contribution of solvent. Incorporating the effect of solvent implicitly or explicitly into protein energy function is necessary. The original version of CHARMM have not consider the factor of solvent. Therefore, many extensions of CHARMM incorporating the influence of solvent have been proposed in the literature, such as SCPISM model [102], EEF1 model [101] and COSMO model [106]. The SASA is the accessible surface to solvent of a biological molecule. It can determine how a biological molecule interact with other molecules, and the protein folding process is influenced by it. The work of Hartlmüller indicates that the utilization of the SASA related information directly is beneficial for FM [107]. We take SASA as the third objective function, which can reflect the effect of solvent implicitly. This treatment also follows the idea that solvation free energy of a protein is proportional to its SASA [96].

Chapter 4

MO3: Incorporation of solvent effect into multi-objective evolutionary strategy

4.1 Method

This section presents a general framework of the proposed approach MO3. Then, the specific components required in the approach to solve PSP are described.

Algorithm 1: Genetic framework of evolutionary strategies

```
begin
    Set the generation counter t=0.
    Generate a population of solutions P.
    Evaluate the fitness of P.
    while Stopping criterion is not met do
        Create offspring population P' by mutating or recombining solutions in
          P.
        Evaluate the fitness of P∪P'.
        Select new population as P.
        t=t+1.
```

The evolution strategy (ES) algorithm was first proposed by Rechenberg in the 1960s [108] and further explored by Schwefel [109]. A generic framework of an implementation of an ES is given in Algorithm 1. It uses the following main components: initiation, mutation, recombination, evaluation and selection. When incorporating multi-objective functions as the fitness function, the framework can be transformed as a framework of a multi-objective evolutionary algorithm. We use the framework to design the multi-objective evolutionary algorithm and reference the well-known evolutionary algorithm, i.e., the Pareto archived evolution strategy [110, 111], to constitute the core of the main algorithm.

4.1.1 Main procedure

Algorithm 2: The main procedure of the multi-objective evaluation algorithm

```
begin
    t=0.
    Generate initial solution $c$ randomly.
    Evaluate the solution $c$.
    Add $c$ to $Archive$.
    while Stopping criterion is not met do
        $c_{mutation1} \leftarrow \text{MUTATION1}(c)$.
        $c_{mutation2} \leftarrow \text{MUTATION2}(c)$.
        Evaluate the solution $c_{mutation1}$ and $c_{mutation2}$.
        if $c_{mutation1}$ dominates $c_{mutation2}$ then
            $c_{buffer} \leftarrow c_{mutation1}$.
        else
            if $c_{mutation2}$ dominates $c_{mutation1}$ then
                $c_{buffer} \leftarrow c_{mutation2}$.
            else
                $c_{buffer} \leftarrow \text{LOW\_ENERGY}(c_{mutation1}, c_{mutation2})$.
                Add $\text{HIGH\_ENERGY}(c_{mutation1}, c_{mutation2})$ to $Archive$.
        if $c$ dominates $c_{buffer}$ then
            throw away $c_{buffer}$.
        else
            if $c_{buffer}$ dominates $c$ then
                Add $c_{buffer}$ to $Archive$.
                $c \leftarrow c_{buffer}$.
            else
                Evaluate $c$ and $c_{buffer}$ according to $Archive$.
                if $c_{buffer}$ is better then
                    Add $c_{buffer}$ to $Archive$.
                    $c \leftarrow c_{buffer}$.
        t=t+1.
```

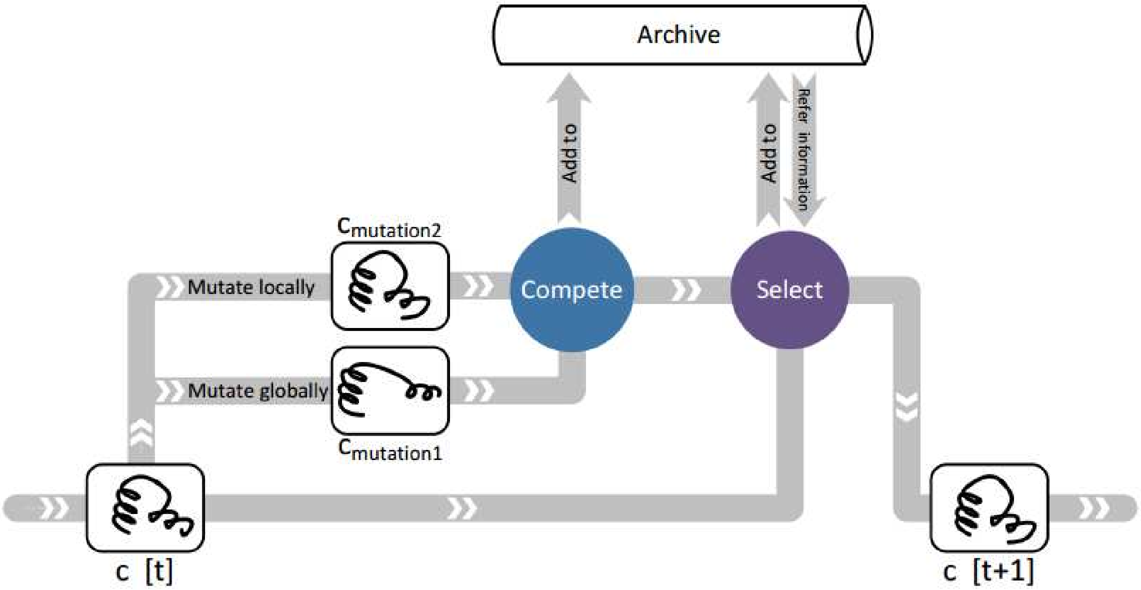


Figure 4.1: Main loop procedure.

Due to space limitations, we primarily describe the main procedure of the algorithm, and the pseudo-code is presented in Algorithm 2.

Initially, the generation counter t is set to 0. A random conformation c is created, and the torsion angles (ϕ, ψ, χ_i) are randomly generated. Subsequently, in the evaluation phase, the bond energy, non-bond energy and SASA of the conformation are evaluated separately by the software tools TINKER and EDTSurf. Later, the solution c is added to the archive of non-dominated solutions *Archive*. From this point, the algorithm starts its main loop. The main procedure is also shown in Fig. 4.1. First, two mutated solutions ($c_{mutation1}$ and $c_{mutation2}$) are generated from current solution c with different mutation operators. They will compete for survival. After evaluation, the better solution is selected as new mutated solution c_{buffer}, while the other one is added to the archive of non-dominated solutions *Archive* if matching the joined conditions. Next, current solution c and mutated solution c_{buffer} are compared for dominance. If one dominates the other, then the dominated one is selected as the new current solution and the other one is discarded. If neither dominates the other, then we use the implied information in *Archive* to evaluate which is better. First, compare c_{buffer} with every one in *Archive*, and discard it if it is dominated by any in *Archive*. Specifically, it is not required to compare c with *Archive* because c is always a non-dominated solution during the main loop process. When c_{buffer} is not dominated by *Archive*, we determine which remains in the least crowded region of the solution space of *Archive*. Finally, the algorithm terminates when the preset number of iterations is reached.

4.1.2 Approximation to Pareto Front

As mentioned above, an archive maintaining a set of non-dominated solutions is created. It serves two purposes in the main procedure: a) store and update all of non-dominated solutions, and b) offer assistance in determining which is better between two solutions in the selection phase. The size of *Archive* is restricted. At each iteration, a solution c^* can be added to *Archive* if

1. *Archive* is empty.
2. *Archive* is not full and c^* is not dominated by any in *Archive*.
3. c^* dominates any solution in *Archive*.
4. *Archive* is full but c^* is non-dominated and in a less crowded space than at least one solution.

In the proposed approach, Pareto dominance selection works to promote convergence by favoring the solutions closer to the Pareto front. The diversity among the non-dominated solutions of *Archive* is ensured by favoring the solutions in a less crowded space. The degree of crowding is calculated by dividing the three-dimensional objective space rigorously in 2^d equal-sized hyper-cubes, where d is defined by the user. Meanwhile, a grid is designed to keep track of crowdedness. Each cell of the grid is a counter to maintain the number of non-dominated solutions residing in each grid location.

4.1.3 Mutation operators

Two types of mutation operators are used. The first operator dramatically changes the conformation by changing all the values of the backbone and side-chain torsion angles of a randomly chosen residue. The second one slightly changes the conformation by perturbing some torsion angles (ϕ, ψ, χ_i) of a randomly chosen residue.

In generic evolutionary strategies, an individual can be represented as a tuple that consists of the decision vector $\mathbf{x}$ and a vector of strategy parameters σ. According to biological observations, offspring are similar to their parents. It is created by adding Gaussian noise to every item of the decision vector $\mathbf{x}$ [112]. The range of the Gaussian noise is controlled by the strategy parameter σ. Therefore, we mutate the protein conformation by changing the angles, which can be described as:

$$\varphi_{new} = \varphi_{old} + \lambda\xi, \tag{4.1}$$

where φ_{new} is the torsion angle after mutating and φ_{old} is the one before mutating. λ is the strategy parameter, and set to 1 in our procedure. ξ is a number that fits a Gaussian distribution of mean $\mu = 0$ and standard deviation $\sigma = 1$.

As suggested in [113], it is more reasonable and plausible to mutate more angles in a protein folding process. In contrast to the mutation rate in [92], we set the probability of the first mutation operator as:

$$M_1 = e^{-\frac{E}{4T_{max}}}, \tag{4.2}$$

where T_{max} is the maximum allowed number of evaluations and E is the number of evaluations performed. For the second mutation operator, the number of mutations is processed as:

$$M_2 = 1 + (\frac{L}{4})e^{-\frac{E}{4T_{max}}}, \tag{4.3}$$

where L is the number of residues. As the number of iterations increases, the probabilities of the first and second mutation operators decrease. Thus, the number of mutations decreases as the search method proceeds.

4.1.4 Computational complexity

We analyze the time complexity of the proposed algorithm as follows.

1. The complexity of the initiation procedure is O(1).

2. The dominance comparison of two solutions needs O(k) time, where k is the number of objective functions.

3. Considering a common iteration in the main loop, L is the length of *Archive*. The procedure of generating and evaluating the mutated solutions needs O(1). In the worst case, after competing, one solution is needed to be added to *Archive*, and it is compared with every other solution in *Archive*. This requires O(kL) comparisons.

4. The later procedure of selecting a better solution from c and c_{buffer} needs O(kL) comparisons in the worst case.

Thus, the complexity of the main loop can be calculated as 2O(kL). This algorithm is continued until the stopping criterion is met. If it ends with N loops, then the overall complexity can be calculated as follows:

$$\begin{aligned} O(1) + N(O(1) + 2O(kL)) = \\ 2O(kNL) + O(N) + O(1). \end{aligned} \tag{4.4}$$

Thus, the approach has time complexity O(kNL). In fact, its largest time expense is close to the calculation of the CHARMM energy and SASA. Because the conformation of a protein is very complex, it can take more than 90% of the running time.

4.1.5 Selecting the Representative Structures

As we can see, the outcome of the proposed algorithm is a set of structures. For a "real" prediction, a method using hierarchical clustering [114, 115] for selecting representative structures is developed. It is shown as the following steps.

1. calculate the RMSD between every two structures of the outcome set (Pareto optimal set) and take the RMSD as the distance metric to measure how close between two structures.

2. use hierarchical clustering to cluster these structures. Repeat combining the closest structures or nodes to form a new node, until a hierarchical tree of binary clusters is built. As a note, Complete linkage clustering is used as linkage criteria.

3. cut the hierarchical tree at a given threshold 4 Å, which define the allowed maximum RMSD between two nodes or structure for joining. N clusters with different size are generated after cutting process.

4. select the centroid of each cluster, which has the lowest average RMSD to all other structure of the cluster.

4.2 Experiment and discussion

4.2.1 Prediction results of four proteins

We have applied our approach to four protein sequences that can be accessed in the Protein Data Bank (PDB). Their PDB IDs are 1ZDD, 1E0M, 1ROP and 1CRN. We run the algorithm at the maximum number of iterations 2×10^4.

Disulfide-stabilized mini protein a domain (PDB id: 1ZDD) consists of 34 residues and 2 helices. Fig. 4.2 shows the Pareto front of 1ZDD computed with the proposed algorithm. As shown in this figure, these solutions are dense and grouped into several clusters. Note that the gray surface in this figure is not the surface of the Pareto front. The surface can help us easily view the three-dimensional Pareto front in the two-dimensional plane.

The Prototype WW domain (PDB ID: 1E0M) consists of 37 residues. It has a triple-stranded antiparallel β-sheet. For this protein, we use the secondary structure information, which is predicted by a protein secondary structure prediction server, PSIPRED We also use the constraints for the secondary structure mentioned above. Fig. 4.3 shows the Pareto front of 1E0M computed with the proposed algorithm.

The COLE1 ROP protein (PDB ID: 1ROP) is a dimer, and each monomer consists almost completely of two alpha helices . 1ROP is composed of 56 residues and forms

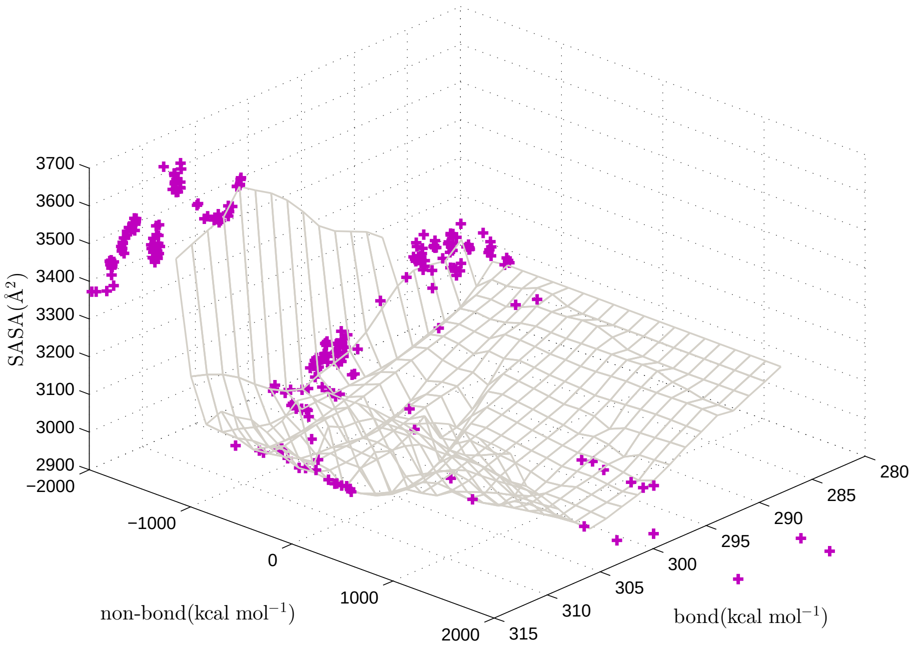


Figure 4.2: Pareto front for 1ZDD protein.

an α-turn secondary structure. Fig. 4.4 shows the Pareto front of 1ROP computed using the proposed algorithm.

Crambin (PDB ID: 1CRN) is a protein with 46 residues. It has two alpha-helices, a pair of beta-strands and three disulfide bonds. It is more complex than the previous proteins. Fig. 4.5 shows the Pareto front of 1CRN computed using our algorithm. These solutions are sparse and mainly clustered in two regions.

Table 4.1 gives the values of thee-objectives (i.e., bond energy, non-bond energy, and SASA) and the RMSD$_{C_\alpha}$ to the target protein for the native, and four non-dominated solutions in *Archive*, respectively. These four non-dominated solutions include the centroid of the cluster with the maximum cluster size, one with the minimum CHARMM22 value, one with the minimum SASA, and one with the minimum RMSD$_{C_\alpha}$. It is clear that solutions with the minimum CHARMM22 or SASA are extreme values located at the Pareto front, and the results in Table 2 suggest that the cluster centroid determined by our decision-making procedure can always perform better. For example, the RMSD$_{C_\alpha}$ of solutions with the minimum CHARMM22 and SASA for 1ZDD are 6.13 and 6.35, respectively. A better solution with RMSD$_{C_\alpha}$ of

Table 4.1: Protein structure prediction results.

Protein	Sequence length	Structure class		bond energy (kcal mol^{-1})	non-bond energy (kcal mol^{-1})	SASA(Å)	RMSD$_{C_\alpha}$(Å)
1ZDD	34	α	native	270.70	-1426.06	3264.06	-
			centroid of cluster	301.28	-1198.88	3091.98	3.26
			min CHARMM22	318.09	-1347.46	3604.93	6.13
			min SASA	340.96	3.00E13	2424.92	6.35
			min RMSD$_{C_\alpha}$	303.99	3.10E5	2940.86	2.16
1E0M	37	β	native	434.53	-39.20	3402.00	-
			centroid of cluster	393.98	-189.44	3955.05	8.00
			min CHARMM22	395.05	-252.34	4572.92	14.62
			min SASA	418.71	9.17E11	3074.77	8.04
			min RMSD$_{C_\alpha}$	399.13	5.29E14	3407.60	5.69
1ROP	56	α	native	-	-	4636.22	-
			centroid of cluster	514.38	4.16E7	3856.07	3.22
			min CHARMM22	512.78	-579.49	5761.62	5.23
			min SASA	528.58	2.68E12	3395.16	4.38
			min RMSD$_{C_\alpha}$	531.84	8.34E6	3935.09	3.07
1CRN	46	$\alpha + \beta$	native	570.17	-712.67	3198.66	-
			centroid of cluster	473.40	8508.00	3323.59	5.56
			min CHARMM22	497.70	363.91	4316.96	6.91
			min SASA	570.90	1.82E15	2702.23	7.83
			min RMSD$_{C_\alpha}$	467.21	1.12E5	3379.44	5.34

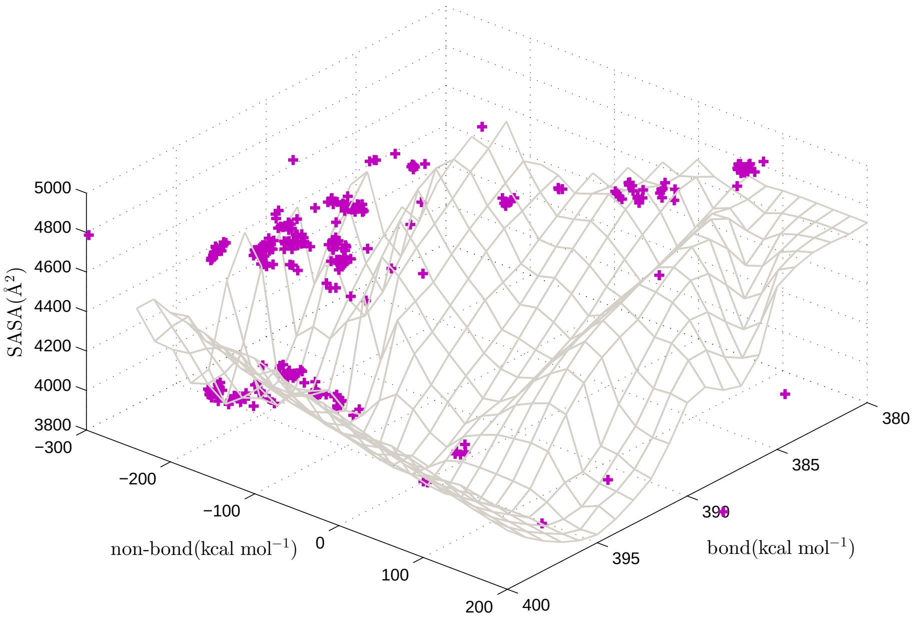


Figure 4.3: Pareto front for 1E0M protein.

3.26 is obtained after our decision-making procedure. Fig. 6 depicts the superpositions of the native and predicted structures for these four proteins.

4.2.2 Verifying whether SASA works

In order to make the energy function more realistic, we have included SASA as the third objective function. We should verify whether this approach works. We have conducted a contrast experiment with two objectives, bond energy and non-bond energy, by employing the same multi-objective evolutionary algorithm. Because the outcome of the multi-objective optimization algorithm is an approximation of the Pareto optimal set, we evaluate the quality of these outcomes to compare performance.

In contrast to a single-objective optimization, the situation of comparison in multi-objective is more complex. The concept of Pareto dominance can be used for comparing two solutions. Moreover, comparing two sets of solutions becomes more complex. As suggested in [116], the quality of an approximation to the Pareto optimal set refers to its convergence and diversity. However, in PSP, the true Pareto optimal front is unknown. Moreover, with different numbers of objectives, the true Pareto

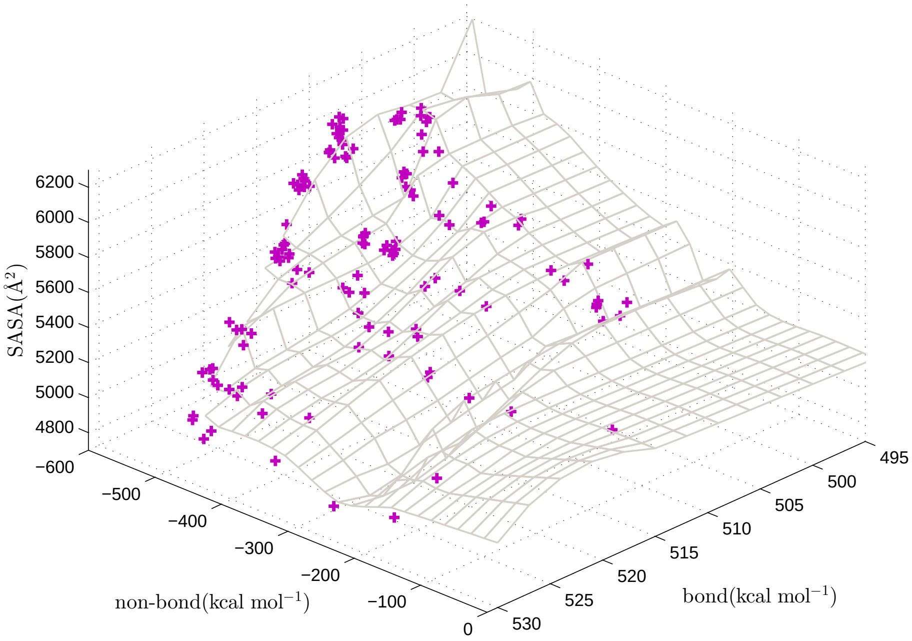


Figure 4.4: Pareto front for 1ROP protein.

optimal front is different. We cannot compare these approximation sets directly. An alternative comparison method is necessary.

Considering our original target, predicting the three-dimensional structure of a protein from its amino acid sequence, we care more about the prediction accuracy. We calculate the RMSD of each solution in the approximation sets. We call the algorithm with two and three objective functions as MO2 and MO3, respectively. The cumulative distribution plot of each set is used to compare the sets. We set the maximum number of iterations to 4×10^4 for MO2. The outcomes of MO2 with the maximum iteration counts 1×10^4 and 4×10^4 are plotted. Those of MO3 with the maximum iteration counts 1×10^4 and 2×10^4 for MO3 are plotted as well. Fig. 4.8 shows the cumulative distribution of RMSD of each set, where t is the number of iterations.

As shown in Fig. 4.8, for MO2, the approximation set of 4×10^4 iterations is superior to the approximation set of 1×10^4 iterations. We can state that the approximation set of 4×10^4 iterations is more accurate (or convergent) than that of

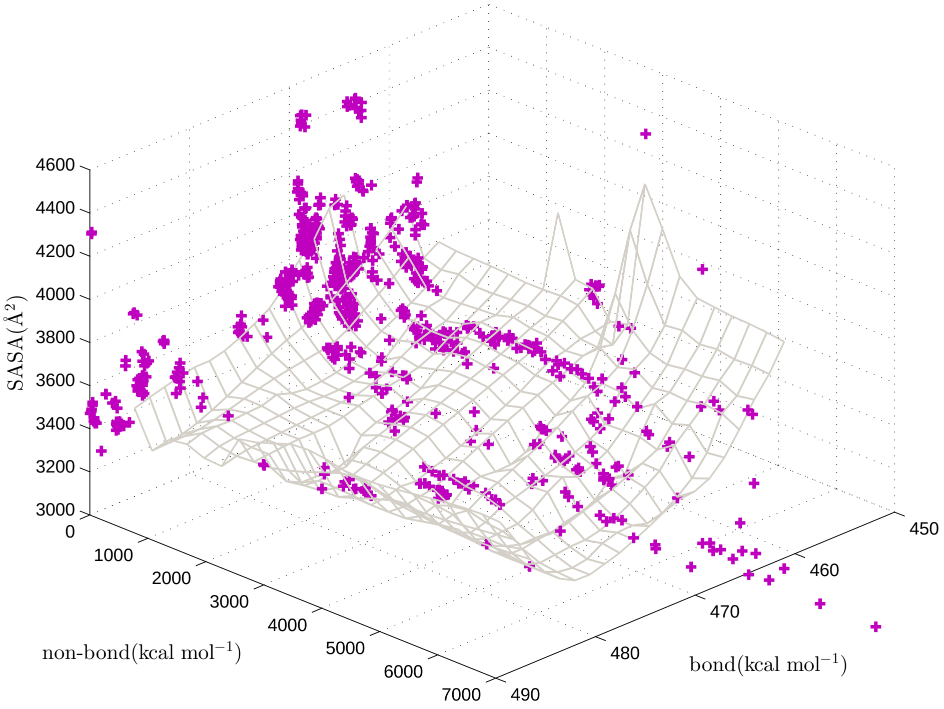


Figure 4.5: Pareto front for 1CRN protein.

1×10^4. For MO3, the approximation set of 1×10^4 iterations is almost as good as the approximation set of 2×10^4 iterations. In other words, MO3 has faster convergence ability than MO2. We obtain the outcomes of MO2 with 4×10^4 iterations and those of MO3 with 2×10^4 iterations as the final results. It is clear that the former is worse than the latter. Meanwhile, the values of MO2 solutions concentrate in a certain range in contrast to a line as in the case of MO3 solutions. This result suggests that the solutions of the approximation set from MO3 is more diverse. Finally, note that Fig. 4.8 is the cumulative distribution plot, not the absolute quantity plot.

Specifically, we analyze the relationship between RMSD (reflecting the accuracy) and SASA. We compare the outcomes of MO2 with 4×10^4 iterations and those of MO3 with 2×10^4 iterations. Fig. 4.7 shows RMSD versus SASA of these four proteins, where the vertical line is the value of SASA of the native protein structures. In MO3, the best solutions concentrate near the value of SASA of the native protein

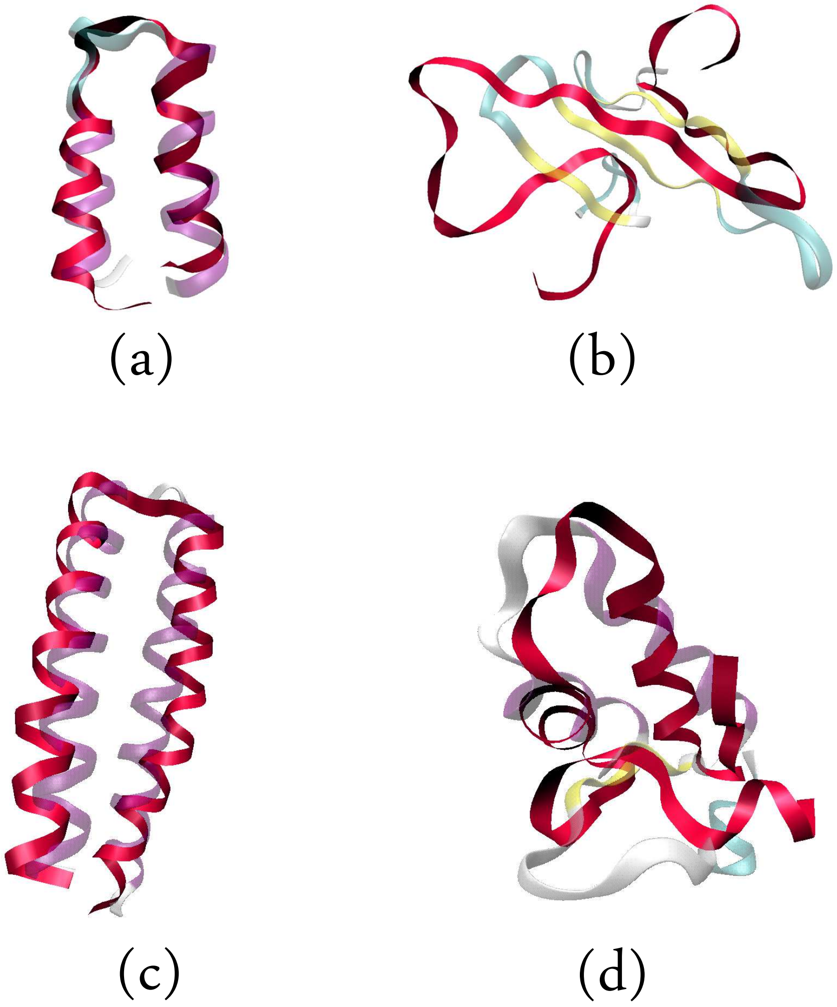


Figure 4.6: Superposition of the native and the predicted structure. a)1ZDD, $\text{RMSD}_{C_\alpha} = 3.26\text{Å}$, b)1E0M, $\text{RMSD}_{C_\alpha} = 8.00\text{Å}$, c)1ROP, $\text{RMSD}_{C_\alpha} = 3.22\text{Å}$, d)1CRN, $\text{RMSD}_{C_\alpha} = 5.56\text{Å}$.

structures. These solutions are grouped into several clusters and are more diverse than those of MO2. It is clear that the smaller SASA, the better conformation of a protein in individual clusters. However, when SASA is smaller than the native value, the situation worsens. The smaller SASA, the worse conformation. In MO2, the solutions all remain on the right side of the vertical line. This result suggests that the effect of solvent is ignored in the evolutionary process. No obvious trend can be found. Note that Fig. 4.7 also shows the number of solutions of different outcomes. It is clear that the number of good solutions of MO3 is far greater than that of MO2.

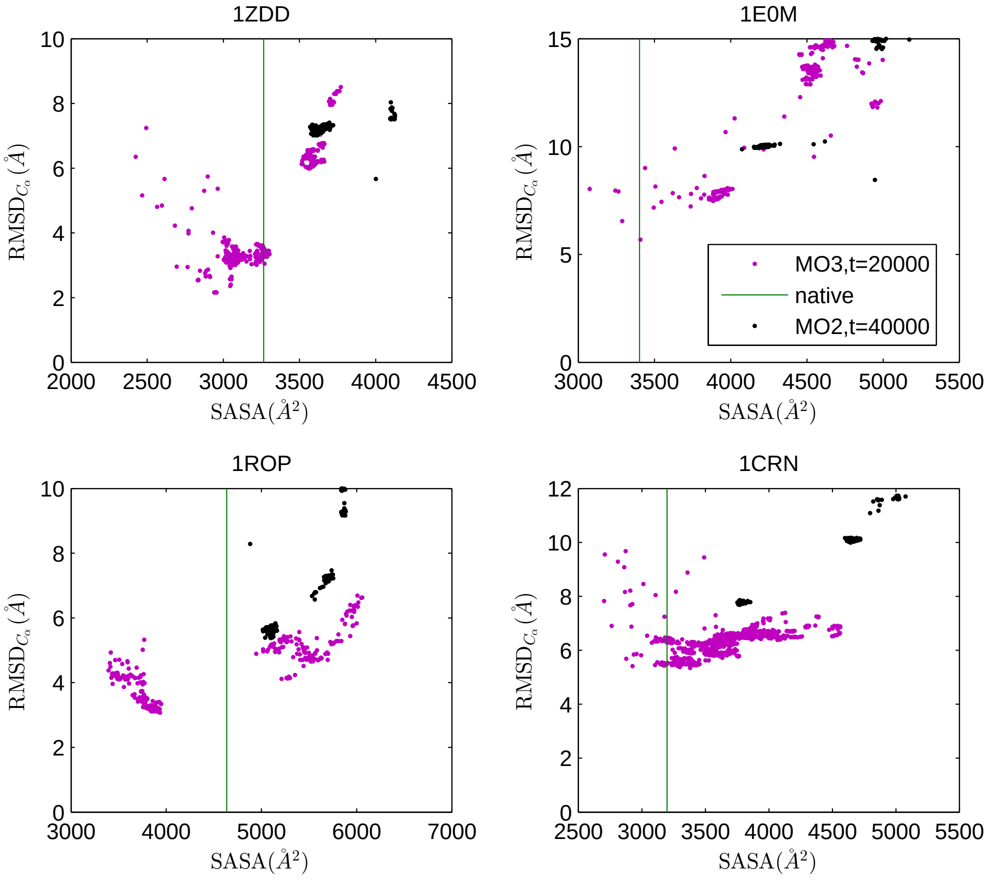


Figure 4.7: RMSD versus SASA of Pareto front for four proteins.

4.2.3 Conflict among the three objectives

As mentioned above, a MOOP is an optimization problem that involves more than one objective to be optimized simultaneously. Coello [88] emphasizes that it is the normal case the objectives of the MOOP are in conflict with each other. It is the typical characteristic of a MOOP. However, there is no formal definition of conflicting objectives in the MOOP field. Generally, a relationship in which the performance of one objective deteriorates as the performance of another improves is considered as a conflict. We use a qualitative method, parallel coordinates plot [117], to identify the conflicting relationship experimentally regarding the set of Pareto optimal solutions.

Fig. 4.9 shows the parallel coordinates plot for the solutions of the protein 1ZDD with three objectives. Similar results have also been obtained for 1E0M, 1ROP, and

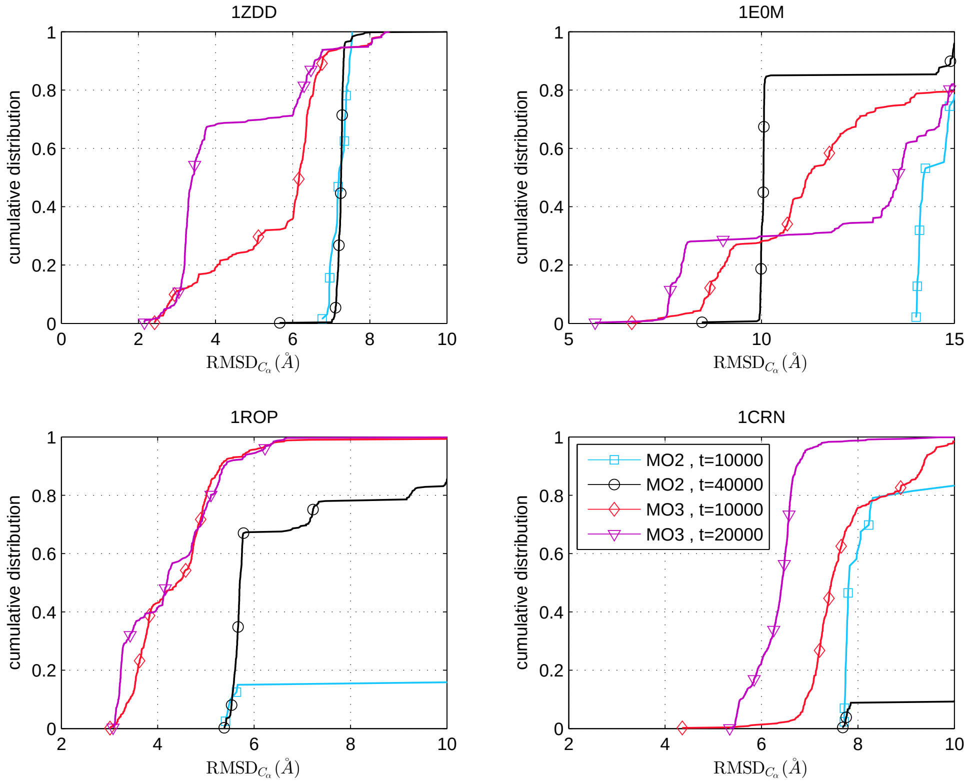


Figure 4.8: Cumulative distribution of RMSD of each set for four proteins.

1CRN. In Fig. 4.9, objective labels are located along the horizontal axis. Normalized values of different objective function values (extreme values have been removed) are indicated on the vertical axis. A solution vector is connected by straight lines. Considering a two objective instance, the line will cross if a conflict is exhibited. Thus, the magnitude of a conflict is visualized as the number of crossing lines. In this figure, It is clear that the three objective functions are in conflict. Moreover, inspecting the degree of three conflicting relationships, SASA is strongly in conflict with the other two objectives.

4.2.4 Comparison with prior work

We have compared our algorithm and results with the other work in the literature. The specific details of the compared methods are given as follows:

1) HC-GA [124] is a hill-climbing genetic algorithm for simulation of protein fold-

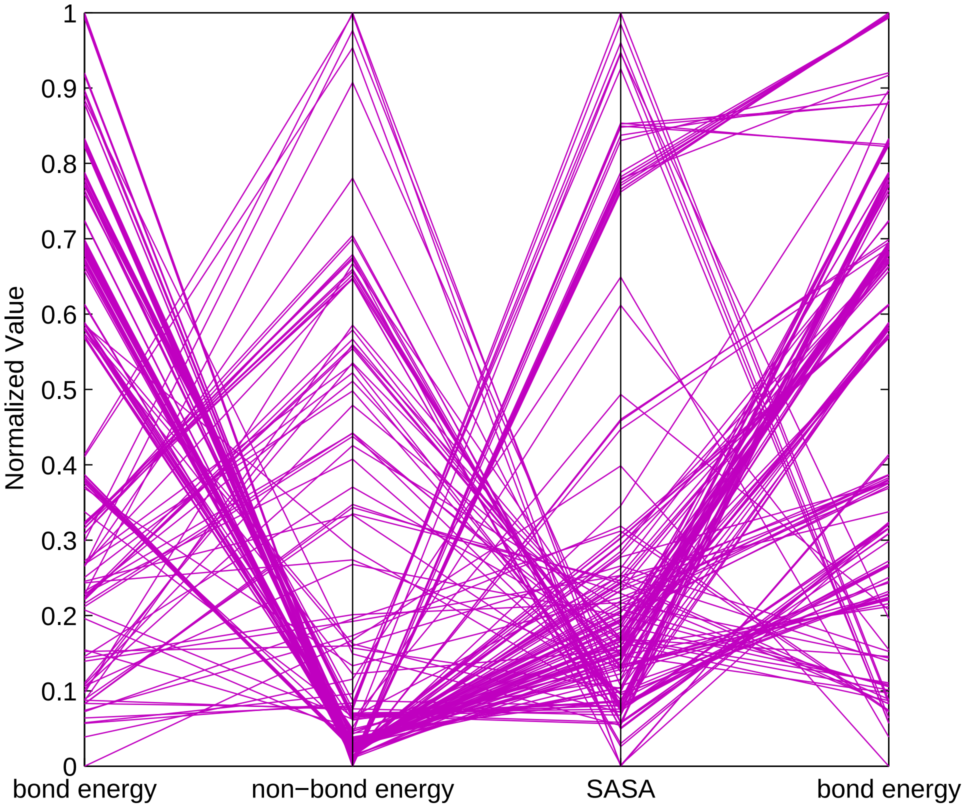


Figure 4.9: Parallel coordinates plot for the solutions of protein 1ZDD (corresponding to Fig. 4.2).

ing which uses a single-objective function as the fitness measurement.

2) I-PAES [92] is the first attempt to use multi-objective functions in an evolutionary approach to perform PSP, and minimizes only two interaction energies, i.e., bond and non-bond energies as objectives.

3) Bhageerat [123] is a PSP software suite for narrowing down the search space of tertiary structures of small proteins. It uses eight different computational modules and can return 10 predictions for a given protein query sequence.

4) NOMAD-PSP [121] uses two direct search algorithms (generalized pattern search and mesh adaptive direct search) to find the optimal solution of PSP that is formalized as a non-linear single-objective optimization problem.

5) IMMALG-DIRECT [122] is a hybrid method that combines an immune al-

Table 4.2: performance comparison with other approaches for the 1ZDD, 1E0M, 1ROP and 1CRN proteins.

Method	Protein	Fra.[a]	Sec.[b]	RMSD	bRMSD[c]
PROPOSED	1ZDD	no	yes	3.26	2.16
	1E0M	no	yes	8.00	5.69
	1ROP	no	yes	3.22	3.07
	1CRN	no	yes	5.56	5.34
ADEMO/D (2016) [118]	1ZDD	no	yes	2.14	-
	1ROP	no	yes	3.83	-
	1CRN	no	yes	6.06	-
GA-WithAPL (2015) [71]	1ZDD	no	yes	-	4.6
	1ROP	no	yes	-	9.8
	1CRN	no	yes	-	5.8
PSO-WithAPL (2015) [71]	1ZDD	no	yes	-	7.2
	1ROP	no	yes	-	10.1
	1CRN	no	yes	-	8.9
HGA (2011) [119]	1ZDD	yes	yes	3.92	-
Parallel framework of NSGA-II (2010) [94]	1ROP	no	yes	3.78	3.39
MI-PAES (2009) [95]	1ZDD	no	yes	-	2.15
	1ROP	no	yes	-	3.48
	1CRN	no	yes	**4.23**	-
CReF (2008) [120]	1ZDD	yes	no	3.4	-
	1ROP	yes	no	7.1	-
NOMAD-PSP (2007) [121]	1ZDD	no	yes	3.87	-
IMMALG-DIRECT (2007) [122]	1ROP	no	yes	3.59	-
I-PAES (2006, 2008) [92] [25]	1ZDD	no	yes	2.27	2.22
	1E0M	no	yes	-	7.27
	1ROP	no	yes	**3.70**	3.50
	1CRN	no	yes	4.43	4.38
Bhageerath (2006) [123]	1ROP	no	yes	4.3	- -
HC-GA (2003) [124]	1CRN	no	yes	5.6	-

[a] whether use fragment-assembly

[b] whether use the secondary structure information

[c] the best RMSD in predicated decoy structures.

gorithm with a quasi-Newton method, aiming to find the lowest CHARMM energy conformation of a given protein sequence.

6) CReF [120] is a central residue fragment based method that makes no use of

entire fragments, but only the phi and psi torsion angle information of the central residue in the template fragments obtained from PDB.

7) MI-PAES [95] develops I-PAES by effectively exploiting some prior knowledge about the hydrophobic interactions.

8) Parallel framework of NSGA-II [94] utilizes an island model of the two-objective evolutionary algorithm to solve PSP.

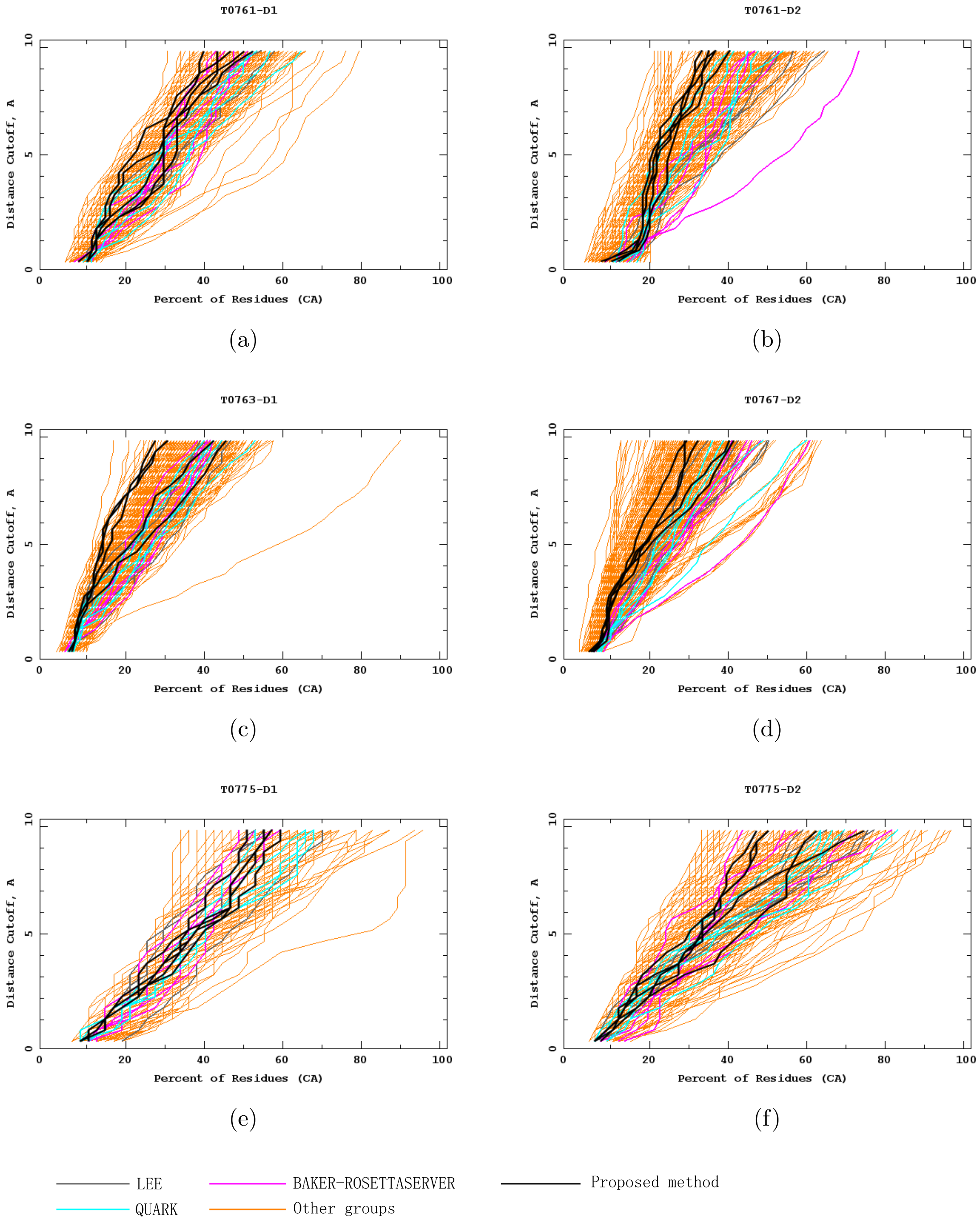
9) HGA [119] improves HC-GA [124] by combining a structured population and a path-relinking procedure to alleviate the local minima trapping problem in genetic algorithms.

10) PSO-WithAPL and GA-WithAPL [71] are angle probability knowledge-based prediction methods based on a genetic algorithm and particle swarm optimization, respectively.

11) ADEMO/D [118] is an adaptive differential evolution algorithm to solve the bond and non-bond energies based two-objective PSP.

Table 4.2 reports the comparison results of our method with other approaches for 1ZDD, 1ROP, and 1CRN proteins. For each of these proteins, a structure within 1-4 Å RMSD (i.e., the root mean square deviation) of the native has been obtained by our method. Inspecting the reported results, our algorithm well outperforms other approaches in terms of the root mean square deviation and is more powerful to solve PSP.

Then, a detailed comparison is carried out among GA-APL [71], the proposed method (i.e., MO3) and its variant MO2 which uses two objective functions on twenty proteins. Table 4.3 records the PDB IDs of proteins, the number of residues which is shown in the bracket, and RMSD results of these three methods. In Table 4 We can observe listed that our proposed method MO3 outperforms MO2, and GA-APL in terms of RMSD.



(a)

(b)

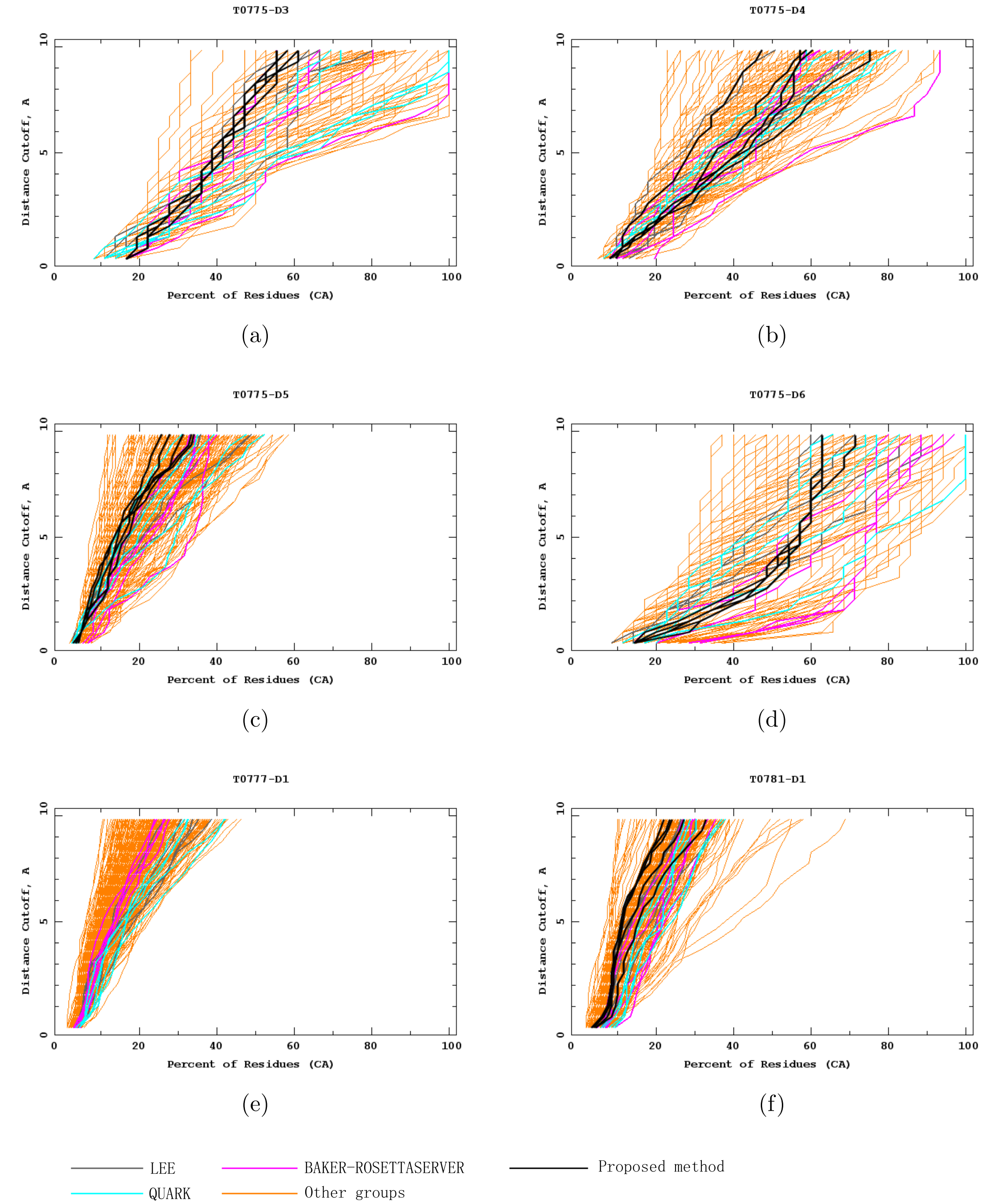
(c)

(d)

(e)

(f)

Figure 4.10: GDT analysis plot for FM targets (a)T0761-D1, (b)T0761-D2, (c)T0763-D1, (d)T0767-D2, (e)T0775-D1, (f)T0775-D2 in CASP11.



(a)

(b)

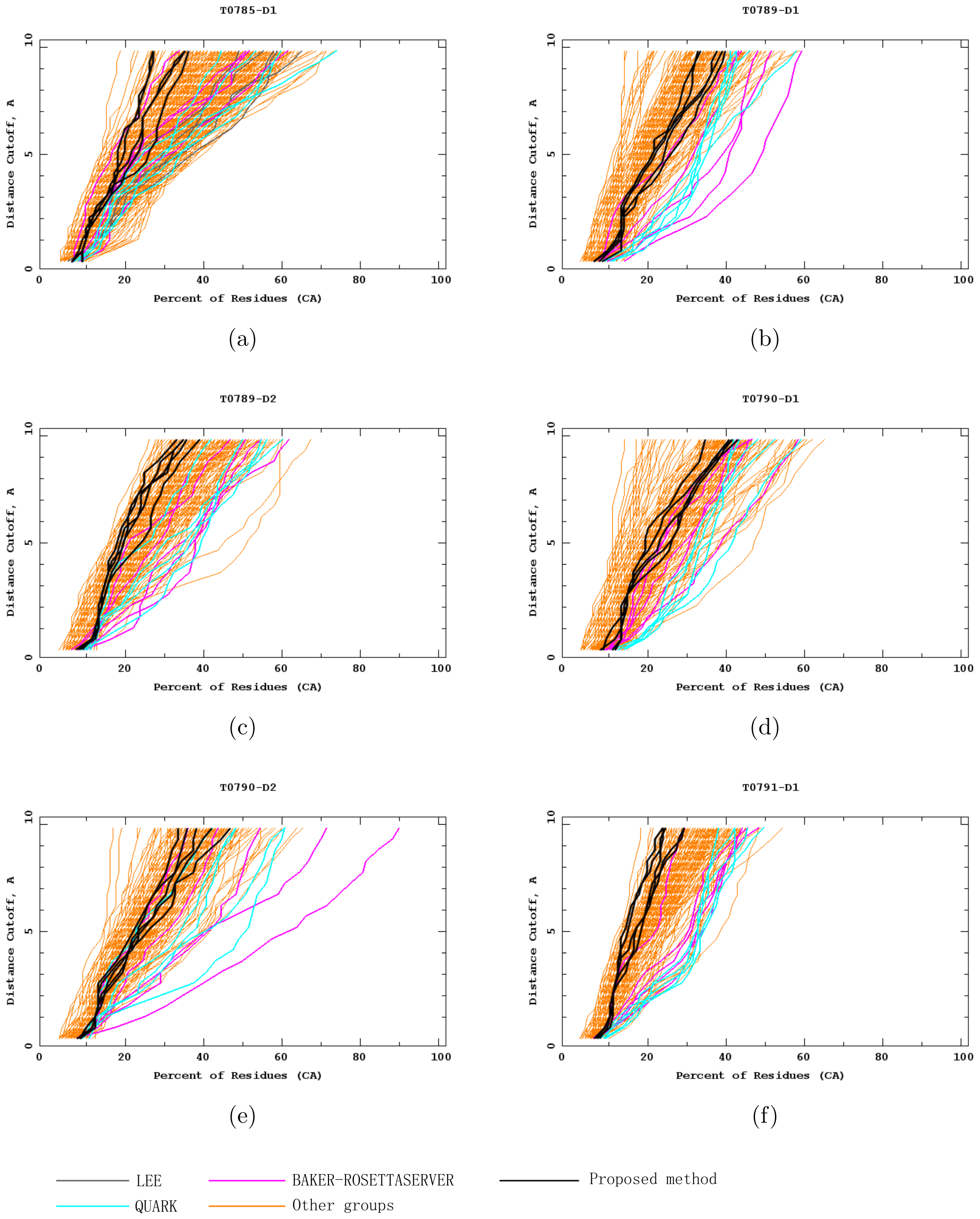
(c)

(d)

(e)

(f)

Figure 4.11: GDT analysis plot for FM targets (a)T0775-D3, (b)T0775-D4, (c)T0775-D5, (d)T0775-D6, (e)T0777-D1, (f)T0781-D1 in CASP11.



(a)

(b)

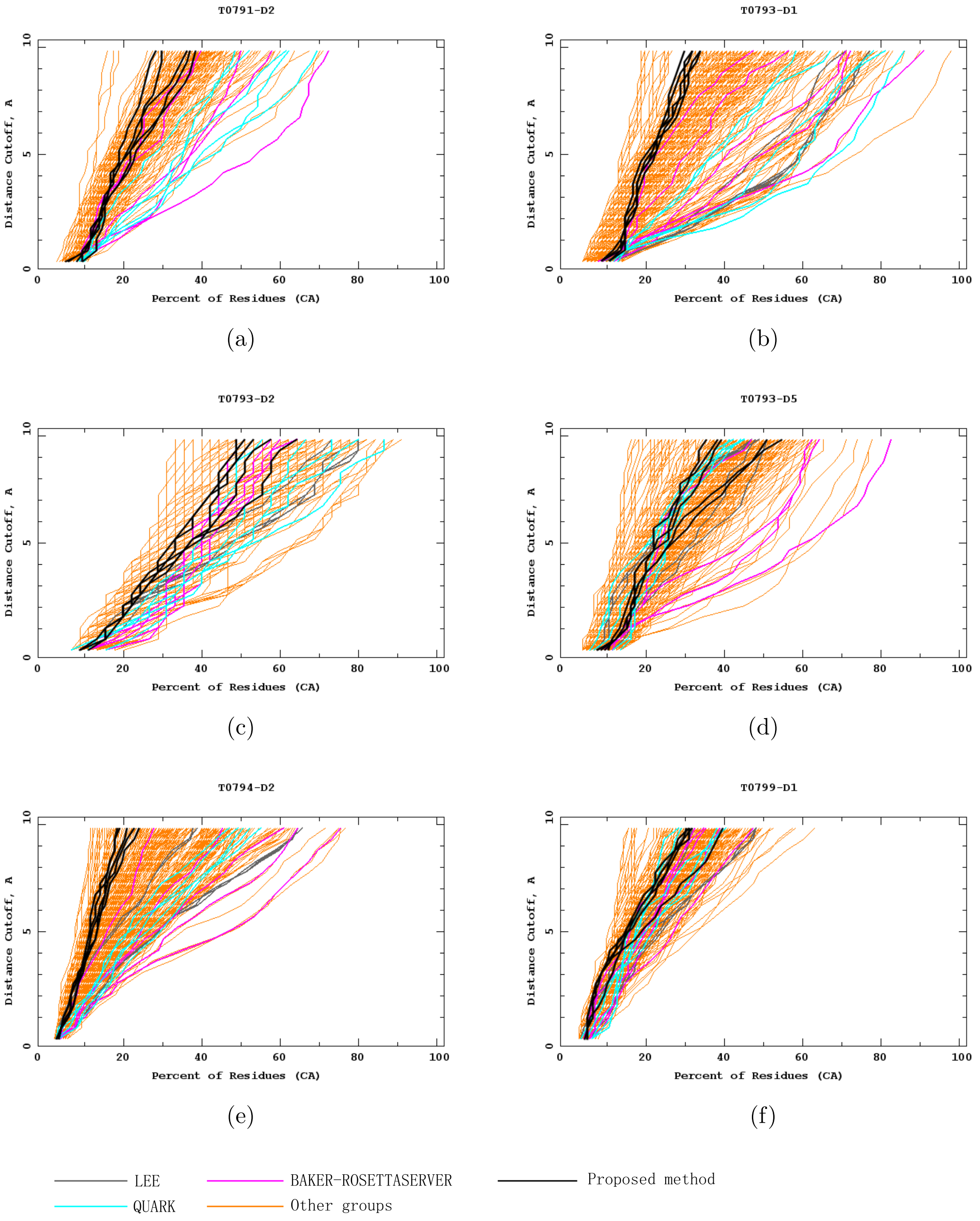
(c)

(d)

(e)

(f)

Figure 4.12: GDT analysis plot for FM targets (a)T0785-D1, (b)T0789-D1, (c)T0789-D2, (d)T0790-D1, (e)T0790-D2, (f)T0791-D1 in CASP11.



(a)

(b)

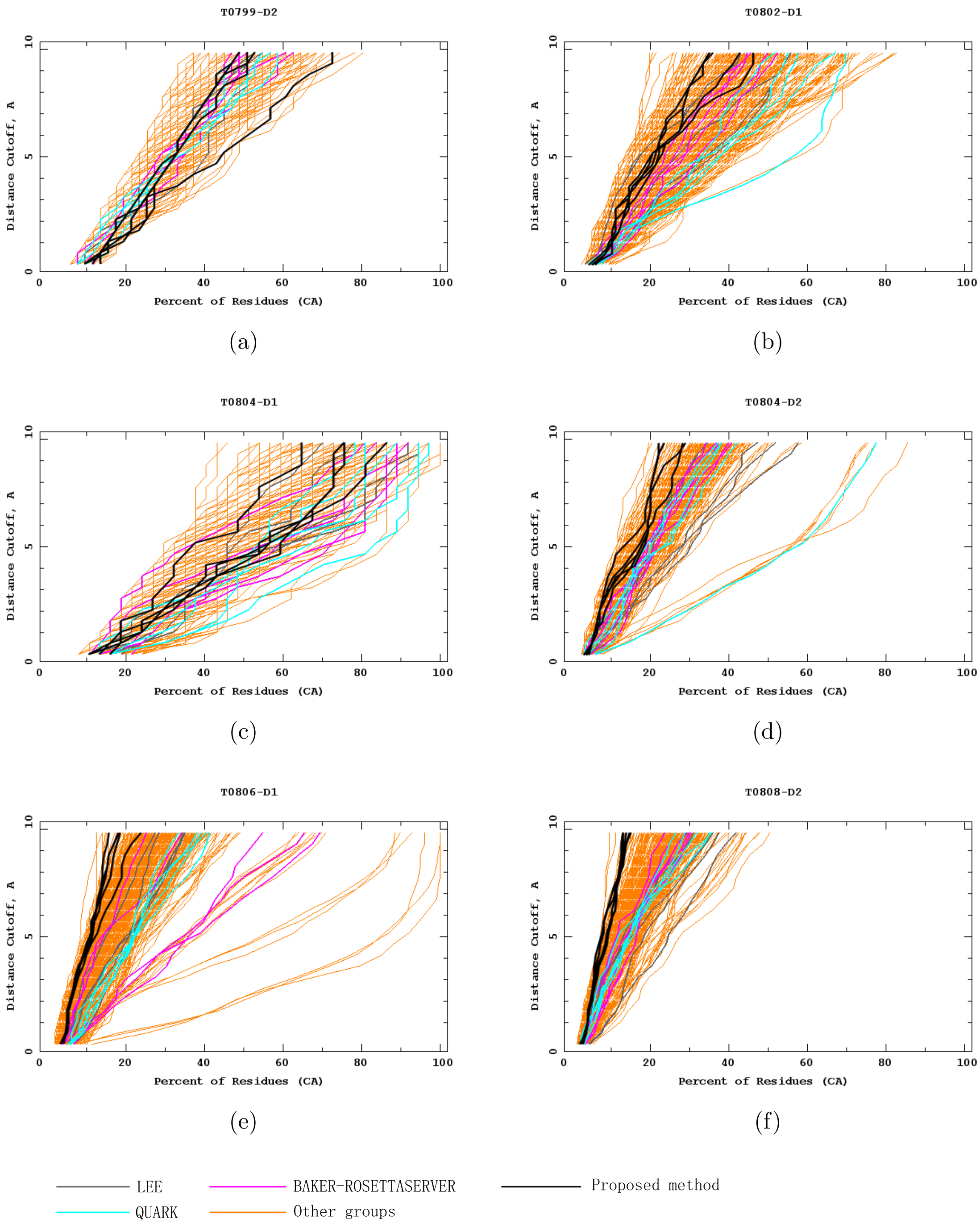
(c)

(d)

(e)

(f)

Figure 4.13: GDT analysis plot for FM targets (a)T0791-D2, (b)T0793-D1, (c)T0793-D2, (d)T0793-D5, (e)T0794-D2, (f)T0799-D1 in CASP11.



(a)

(b)

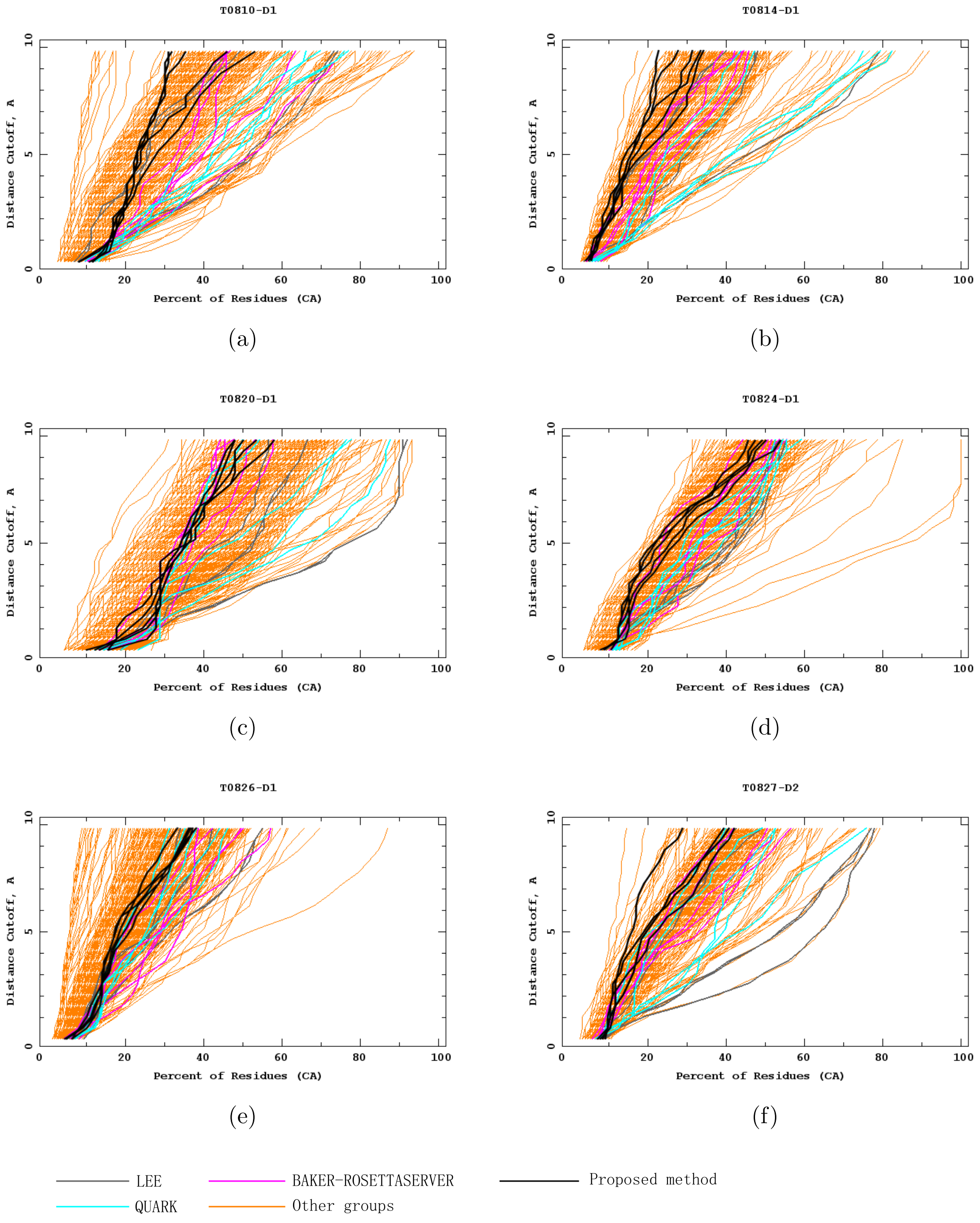
(c)

(d)

(e)

(f)

Figure 4.14: GDT analysis plot for FM targets (a)T0799-D2, (b)T0802-D1, (c)T0804-D1, (d)T0804-D2, (e)T0806-D1, (f)T0808-D2 in CASP11.



(a)

(b)

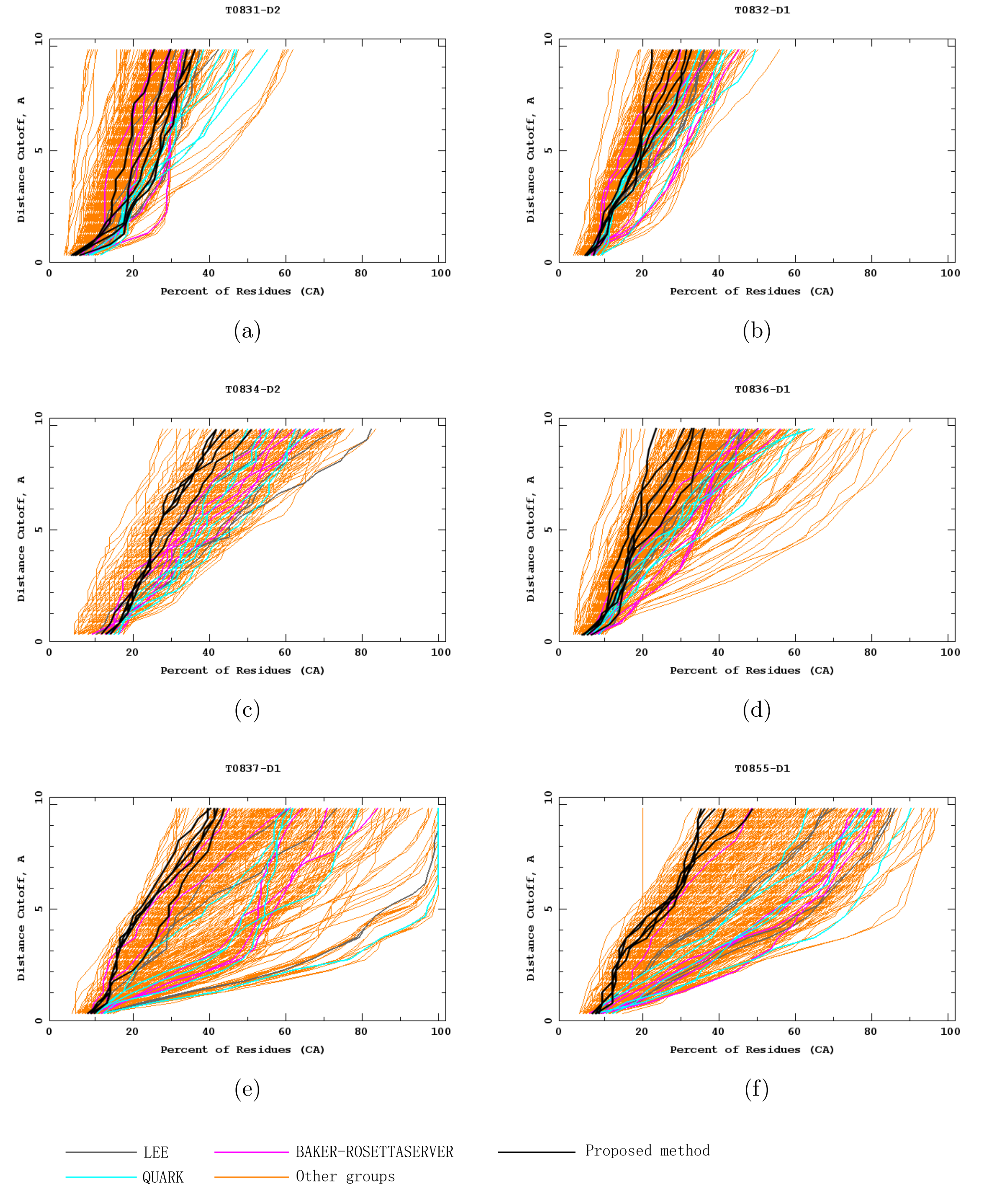
(c)

(d)

(e)

(f)

Figure 4.15: GDT analysis plot for FM targets (a)T0810-D1, (b)T0814-D1, (c)T0820-D1, (d)T0824-D1, (e)T0826-D1, (f)T0827-D2 in CASP11.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.16: GDT analysis plot for FM targets (a)T0831-D2, (b)T0832-D1, (c)T0834-D2, (d)T0836-D1, (e)T0837-D1, (f)T0855-D1 in CASP11.

Table 4.3: Performance Comparison among MO3, MO2 and GA-APL on 20 Proteins.

		RMSD		
Protein	Length	MO3	MO2	GA-APL
3P7K	45	2.02	3.43	2.09
2MTW	20	4.49	4.78	2.48
1WQC	26	4.65	4.56	5.24
2P81	44	4.30	11.89	8.53
1L2Y	20	3.44	5.11	5.28
3V1A	48	2.23	4.70	10.70
2P6J	52	5.96	16.93	15.18
2F4K	33	5.91	5.24	6.60
1ENH	54	11.99	14.31	14.99
2MR9	44	6.68	12.11	9.22
1AIL	70	9.97	16.45	19.57
2PMR	76	10.12	11.82	21.54
2JUC	59	10.80	16.08	18.50
1K43	14	2.86	2.73	3.55
1DFN	30	7.45	10.15	10.21
1D5Q	27	6.71	11.09	6.51
1ACW	29	7.45	9.11	10.66
1Q2K	31	7.93	16.57	7.59
1AB1	46	7.52	10.09	10.10
2P5K	63	9.23	9.95	13.97

4.2.5 Comparing with CASP Competitors

To further verify the performance of the proposed method, all available single-domain FM protein targets up to 345 residues taken from the 11th CASP experiment (CASP11) are tested. Table 4.4 summarizes the GDT-TS values of the top five solutions obtained by our method and three state-of-the-art methods, i.e., LEE [46], QUARK [14], and BAKER-ROSETTASERVER [64] for the targets T0761-D1 and T0799-D2. The GDT-TS values of other tested proteins are in Table 4.5 and Table 4.6. Fig. 4.10 $\sim$ Fig. 4.16 represents the results of GDT analysis for all Test proteins.

From Table 4.4, 4.5, 4.6 we can find that our method is generally capable of finding satisfying solutions for small-size proteins. Promisingly, it can find a better solution (with a GDT-TS value of 31.37) than three compared state-of-the-art methods for the target T0799-D2. It is probably owing to the fact that small proteins have smaller conformational search space, which is easier for the evolutionary search algorithms to

Table 4.4: GDT-TS Results Between Our Proposed Method and Three State-of-the-Art Methods for the Targets T0761-D1 and T0799-D2.

Target (Length)	Group	GDT-TS				
T0761-D1 (88)	LEE	25.85	25.85	25.85	25.85	21.59
	ROSETTA[a]	28.12	24.74	23.31	23.30	22.44
	QUARK	28.12	27.84	24.15	23.86	22.44
	MO3	22.44	22.44	22.15	20.45	19.31
T0799-D2 (51)	LEE	25.98	24.02	23.53	22.55	22.55
	ROSETTA	27.94	24.51	23.53	23.53	23.04
	QUARK	26.96	25.98	25.49	23.53	23.04
	MO3	31.37	26.47	25.98	25.98	25.49

identify the correct fold. As can be seen from Fig. 4.10 ∼ Fig. 4.16, solutions obtained by our proposed method are generally competitive among all CASP11 competitors. To be more specific, we normalize all the GDT-TS values according to their mean and standard deviation to obtain the corresponding Z-score. The cumulative Z-scores shown in Table 4.7, are calculated based on the collected data from the FM website, and our result is based on a randomly selected solution from the top five solutions with largest sizes of the Pareto optimal set. The analysis of Z-scores for GDT-TS shows that our method ranks 54th among all 140 compared groups, while three state-of-the-art methods rank 8th, 17th, and 35th respectively. Although our method is worse than these three state-of-the-art methods, it can be said that our proposed three-objective evolutionary algorithm is a competitive method and performs above average as far as all groups are concerned in this blind test.

Finally, we note that the advantages of the above three state-of-the-art methods are owing to the principle of fragment assembly [46, 14, 64]. In a fragment assembly based method, the query sequence is fragmented into fully overlapping short stretches of amino acids. Conventional template-based techniques are used to generate candidate structures for these small fragments. These structural fragments are then sampled (e.g., using Monte Carlo simulation) and assembled to construct a low-energy protein conformation. The success of these methods depends on the sophisticated fragment generation and conformational movements, thus making the

Table 4.5: Protein structure prediction results for all CASP test proteins (1).

Target (Length)	Group	GDT-TS					Target (Length)	Group	GDT-TS				
T0761-D1 (88)	LEE	25.85	25.85	25.85	25.85	21.59	T0793-D5 (118)	LEE	26.20	23.56	17.07	17.07	16.83
	ROSETTA	28.12	24.72	23.30	23.01	22.44		ROSETTA	40.38	34.13	29.81	29.81	21.15
	QUARK	28.12	27.84	24.15	23.86	22.44		QUARK	22.36	21.64	19.47	15.62	15.14
	MO3	22.44	22.44	22.15	20.45	19.31		MO3	22.35	22.11	20.67	19.23	18.02
T0761-D2 (136)	LEE	28.32	27.66	27.66	27.66	27.21	T0794-D2 (172)	LEE	22.67	22.67	22.38	12.50	12.50
	ROSETTA	38.27	28.98	27.66	24.78	24.56		ROSETTA	27.62	24.86	20.64	18.46	11.77
	QUARK	27.43	26.33	25.89	24.34	19.25		QUARK	20.64	20.06	17.59	17.30	16.28
	MO3	23.23	22.56	21.46	21.23	20.79		MO3	10.02	9.59	9.59	9.59	9.30
T0763-D1 (130)	LEE	20.77	20.39	16.35	16.15	15.00	T0799-D1 (141)	LEE	19.15	18.97	18.97	13.12	12.06
	ROSETTA	20.77	19.42	19.23	17.69	16.35		ROSETTA	19.86	15.96	15.60	14.89	13.12
	QUARK	20.00	19.23	18.65	17.31	16.35		QUARK	16.49	16.14	15.25	14.89	14.36
	MO3	17.88	15.57	12.88	12.88	12.88		MO3	14.89	12.41	12.05	11.70	11.34
T0767-D2 (180)	LEE	21.25	21.11	19.17	18.19	15.97	T0799-D2 (51)	LEE	25.98	24.02	23.53	22.55	22.55
	ROSETTA	28.61	20.97	20.56	19.58	18.47		ROSETTA	27.94	24.51	23.53	23.53	23.04
	QUARK	25.56	19.72	19.31	18.19	17.92		QUARK	26.96	25.98	25.49	23.53	23.04
	MO3	16.94	15.55	15.55	14.58	13.61		MO3	31.37	26.47	25.98	25.98	25.49
T0775-D1 (47)	LEE	38.30	28.19	27.13	23.94	23.94	T0802-D1 (116)	LEE	26.29	25.00	20.69	15.52	15.30
	ROSETTA	31.38	30.85	29.79	27.13	23.40		ROSETTA	23.28	23.06	20.47	19.61	17.67
	QUARK	31.91	31.38	31.38	30.32	28.19		QUARK	33.84	26.94	25.86	24.78	24.57
	MO3	30.31	28.19	27.66	27.12	25.53		MO3	18.31	17.67	16.59	16.37	15.51
T0775-D2 (66)	LEE	31.44	29.92	26.89	26.52	22.73	T0804-D1 (37)	LEE	48.65	45.95	45.27	37.84	36.49
	ROSETTA	35.23	31.82	26.89	25.00	21.21		ROSETTA	50.00	48.65	45.27	37.84	30.41
	QUARK	33.71	28.41	28.03	26.52	25.38		QUARK	58.11	50.68	45.95	43.24	39.19
	MO3	29.16	26.51	23.48	23.48	20.83		MO3	40.54	40.54	39.86	39.18	31.08
T0775-D3 (36)	LEE	37.50	34.03	31.94	31.25	30.56	T0804-D2 (152)	LEE	21.38	21.05	20.56	14.64	14.64
	ROSETTA	52.78	38.19	35.42	34.72	33.33		ROSETTA	17.43	15.95	15.79	14.80	14.47
	QUARK	48.61	45.83	43.75	34.72	30.56		QUARK	38.65	18.26	15.62	15.13	14.80
	MO3	34.02	34.02	33.33	32.63	31.94		MO3	12.50	12.17	11.51	11.51	11.02
T0775-D4 (61)	LEE	34.43	33.61	31.15	26.23	20.49	T0806-D1 (256)	LEE	17.58	14.84	14.75	14.36	11.62
	ROSETTA	45.90	32.79	30.33	27.46	24.59		ROSETTA	26.17	25.78	24.90	13.09	10.94
	QUARK	34.02	30.33	28.69	26.64	26.23		QUARK	17.48	16.99	16.41	16.31	16.21
	MO3	31.96	29.50	28.68	28.27	22.13		MO3	9.76	8.78	8.69	8.59	8.39
T0775-D5 (145)	LEE	16.55	15.17	14.31	12.93	12.41	T0808-D2 (269)	LEE	18.59	15.8	11.71	10.22	9.76
	ROSETTA	21.38	18.28	15.86	15.00	12.76		ROSETTA	12.08	11.9	11.71	11.06	9.76
	QUARK	20.17	17.07	14.14	12.41	11.55		QUARK	12.55	11.24	11.15	10.97	10.78
	MO3	13.10	13.10	12.41	11.89	11.03		MO3	6.87	6.78	6.78	6.59	6.59
T0775-D6 (35)	LEE	42.86	39.29	35.71	33.57	32.86	T0810-D1 (113)	LEE	37.83	36.06	36.06	20.35	20.13
	ROSETTA	68.57	67.14	65.71	54.29	41.43		ROSETTA	36.73	31.86	27.88	26.77	24.56
	QUARK	62.14	45.71	37.14	37.14	33.57		QUARK	33.63	33.63	32.97	32.74	30.97
	MO3	45.71	43.57	42.85	42.14	40.71		MO3	23.00	22.12	20.79	20.57	20.13
T0777-D1 (345)	LEE	14.2	12.61	12.46	11.59	11.45	T0814-D1 (137)	LEE	32.85	24.64	24.64	15.15	14.60
	ROSETTA	11.67	11.38	10.36	10.22	10.22		ROSETTA	23.36	20.62	19.89	17.52	15.33
	QUARK	16.16	13.99	13.99	13.62	12.68		QUARK	33.39	31.39	31.39	22.81	21.53
	MO3	0.01	0.01	0.01	0.01	0.01		MO3	14.41	13.50	12.95	12.59	11.67

Table 4.6: Protein structure prediction results for all CASP test proteins (2).

Target (Length)	Group	GDT-TS					Target (Length)	Group	GDT-TS				
T0781-D1 (200)	LEE	16.00	14.75	12.25	12.12	12.12	T0820-D1 (90)	LEE	53.06	53.06	38.06	38.06	35.83
	ROSETTA	19.25	17.00	15.75	14.00	12.25		ROSETTA	36.11	30.83	30.56	29.72	29.17
	QUARK	18.62	16.75	16.50	15.75	14.75		QUARK	47.22	41.94	38.61	32.78	32.78
	MO3	14.62	12.75	11.50	11.37	11.00		MO3	31.38	30.55	30.27	29.72	28.05
T0785-D1 (112)	LEE	27.45	24.33	23.88	23.21	21.65	T0824-D1 (108)	LEE	30.32	30.32	28.24	27.78	27.78
	ROSETTA	25.67	21.20	20.98	20.54	13.62		ROSETTA	28.47	27.32	25.23	24.54	21.53
	QUARK	27.23	26.12	25.89	21.88	21.88		QUARK	29.17	28.70	27.08	27.08	25.69
	MO3	16.51	16.51	16.07	14.73	14.73		MO3	22.45	21.52	21.29	20.13	19.90
T0789-D1 (146)	LEE	-	-	-	-	-	T0826-D1 (201)	LEE	25.00	25.00	18.41	18.16	18.04
	ROSETTA	36.89	30.94	27.45	23.60	18.36		ROSETTA	23.38	23.26	21.89	21.52	18.28
	QUARK	30.77	27.80	26.92	26.75	25.70		QUARK	22.51	20.40	19.90	19.28	16.29
	MO3	19.93	19.23	18.53	18.35	17.30		MO3	17.28	17.16	16.79	15.92	15.67
T0789-D2 (126)	LEE	-	-	-	-	-	T0827-D2 (158)	LEE	42.50	36.67	36.00	20.00	19.83
	ROSETTA	28.97	28.97	26.98	20.64	20.44		ROSETTA	22.00	21.83	21.17	19.00	16.50
	QUARK	28.57	26.19	25.59	24.21	19.44		QUARK	30.83	28.17	26.17	20.83	17.17
	MO3	18.25	17.06	17.06	16.86	16.46		MO3	17.83	17.00	16.83	16.16	13.66
T0790-D1 (135)	LEE	-	-	-	-	-	T0831-D2 (244)	LEE	22.46	22.21	21.45	20.68	17.26
	ROSETTA	-	-	-	-	-		ROSETTA	25.89	22.34	19.29	17.00	14.21
	QUARK	-	-	-	-	-		QUARK	25.51	25.51	23.86	23.35	22.46
	MO3	20.55	20.00	19.44	18.88	18.14		MO3	22.71	21.44	19.54	18.40	15.99
T0790-D2 (130)	LEE	-	-	-	-	-	T0832-D1 (209)	LEE	18.06	18.06	17.82	16.99	16.63
	ROSETTA	-	-	-	-	-		ROSETTA	24.04	22.73	19.50	14.83	13.28
	QUARK	-	-	-	-	-		QUARK	24.28	19.86	18.06	17.11	16.87
	MO3	20.57	19.61	18.84	18.65	17.88		MO3	16.62	16.62	15.07	14.83	14.71
T0791-D1 (156)	LEE	-	-	-	-	-	T0834-D2 (92)	LEE	35.47	33.14	31.11	26.16	26.16
	ROSETTA	-	-	-	-	-		ROSETTA	31.69	30.81	28.20	27.91	27.33
	QUARK	-	-	-	-	-		QUARK	34.88	32.27	30.81	29.94	29.36
	MO3	14.09	14.09	13.92	12.91	12.91		MO3	24.41	24.12	24.12	23.83	23.83
T0791-D2 (139)	LEE	-	-	-	-	-	T0836-D1 (204)	LEE	20.83	20.83	20.47	19.98	16.54
	ROSETTA	-	-	-	-	-		ROSETTA	26.96	26.10	25.00	21.81	17.89
	QUARK	-	-	-	-	-		QUARK	25.49	24.27	21.45	20.59	19.85
	MO3	18.65	18.47	17.93	17.02	16.12		MO3	19.36	16.66	16.42	15.31	14.33
T0793-D1 (109)	LEE	42.57	42.57	42.08	41.83	37.87	T0837-D1 (121)	LEE	62.81	62.60	30.37	30.37	26.45
	ROSETTA	43.07	37.13	35.15	24.75	20.79		ROSETTA	43.39	39.67	39.26	25.00	20.04
	QUARK	48.27	40.84	39.60	29.45	29.21		QUARK	65.70	40.50	40.29	34.92	29.34
	MO3	19.05	18.81	18.81	18.31	17.82		MO3	23.55	21.07	20.24	19.62	19.62
T0793-D2 (45)	LEE	35.00	35.00	34.44	34.44	32.78	T0855-D1 (115)	LEE	44.35	43.26	30.43	29.78	29.57
	ROSETTA	36.67	35.00	33.33	32.78	31.11		ROSETTA	46.52	41.09	39.78	39.56	23.04
	QUARK	38.89	35.56	35.00	34.44	31.67		QUARK	50.65	41.52	39.35	34.13	30.43
	MO3	29.44	28.33	27.77	27.77	26.66		MO3	20.21	18.69	18.04	17.82	17.60

procedure highly complicated and difficult to be well-tuned. On the contrary, our method is not based on fragments, but instead based on a simple torsion angles protein representation and a three-objective evolutionary search algorithm. It is more straightforward to predict the 3-D structure of a protein from its one-dimensional sequence of amino acids.

Table 4.7: Cumulative Z-score of GDT-TS score on FM Targets in CASP11.

Group name	SUM Z-score	Rank	Group name	SUM Z-score	Rank
Kiharalab	41.61	1	PhyreX	-15.28	71
Jones-UCL	37.43	2	Legato	-15.37	72
Zhang	33.88	2	BioSerf	-15.53	73
ProQ2	31.37	4	eThread	-15.62	74
SHORTLE	30.55	5	wfKeasar-PTIGRESS	-16.03	75
Seok-refine	29.94	6	FLOUDAS_A2	-16.08	76
MULTICOM	28.91	7	HHPredX	-16.56	77
LEE	**28.71**	**8**	KIAS-GDANSK	-18.77	78
ProQ2-refine	28.28	9	Alpha-Gelly-Server	-19.31	79
Zhang-Server	27.65	10	LmtdSeder	-19.55	80
TASSER	27.50	11	HHPredA	-19.97	81
QA-RecombineIt_H	27.14	12	wfMix-KFb	-20.16	82
Wallner	26.60	13	Seder2	-21.72	83
LEER	25.75	14	2PG	-23.19	84
Skwark	25.52	15	wfHHPred-PTIGRESS	-23.51	85
BAKER	25.33	16	FLOUDAS_A1	-24.69	86
QUARK	**24.65**	**17**	wfAll-MD-RFLB	-25.77	87
QA-RecombineIt_WFH	22.15	18	Bilab	-26.62	88
CNIO	21.62	19	BioShell-server	-26.77	89
RosEda	21.30	20	IntFOLD3	-26.78	90
PML	21.28	21	MUFOLD-Server	-27.11	91
MUFOLD-R	18.20	22	SAM-T08-server	-30.77	92
QA-RecombineIt_H2	17.42	23	slbio	-32.35	93
wfMix-KPa	17.22	24	DELCLAB	-33.82	94
Mufold	15.24	25	chuo-fams-server	-36.13	95
keasar	13.87	26	Atome2_CBS	-37.84	96
McGuffin	12.59	27	wfAll-Cheng	-38.01	97
NEFILIM	12.02	28	raghavagps-tsppred	-39.30	98
wfMix-KPb	11.93	29	chuo-fams	-41.07	99
Boniecki_pred	10.97	30	3D-Jigsaw-V5_1	-42.40	100
Seder1	9.78	31	ALAdeGAP	-43.69	101
RBO_Aleph	9.16	32	WY-C	-44.47	102
nns	8.70	33	wfZhng-Ksr	-48.44	103
wfMix-KFa	5.77	34	wfZhng-Sk-BW	-52.13	104
BAKER-ROSETTASERVER	**5.71**	**35**	Victoria	-52.30	105
FLOUDAS_A4	2.86	36	STAP	-53.01	106
TASSER-VMT	1.00	37	dppred	-54.35	107
myprotein-me	0.68	38	wfKsrFdit-BW-Sk-BW	-56.39	108
MULTICOM-CONSTRUCT	0.53	39	wfKsrFdit-BW-Sk-McG	-56.76	109
MULTICOM-CLUSTER	0.33	40	wf-Void_Crushers	-56.82	110
Gong3701	-0.96	41	wf-AnthropicDreams	-57.11	111
MULTICOM-REFINE	-1.28	42	WeFold-GoScience	-57.96	112
Seok	-1.49	43	Sun_Tsinghua	-58.08	113
MULTICOM-NOVEL	-2.64	44	FFAS03	-58.29	114
RaptorX	-3.38	45	PSF	-58.34	115
Seok-server	-4.83	46	MATRIX	-61.68	116
FUSION	-5.72	47	SSThread	-61.84	117
Handl	-6.19	48	InnoUNRES	-63.84	118
STRINGS	-6.73	49	Rosetta_at_Kingston	-64.06	119
RaptorX-FM	-7.32	50	WeFold-Contenders	-64.52	120
Pcons-net	-7.59	51	OPIG	-65.55	121
FALCON_TOPO	-8.21	52	wf-Baker-UNRES	-66.56	122
FFAS-3D	-8.27	53	CASPITA3D	-67.19	123
MO3 (PROPOSED)	**-8.49**	**54**	Laufer	-72.16	124
ZHOU-SPARKS-X	-8.72	55	LNCCUnB	-72.38	125
Pareto	-9.43	56	WeFold-Wiskers	-73.96	126
BhageerathH	-9.97	57	pkfc	-74.05	127
FALCON_MANUAL	-10.38	58	Void_Crushers	-74.77	128
FALCON_EnvFold	-10.38	59	Mongolian_Team	-74.78	129
FALCON_MANUAL_X	-10.85	60	Contenders	-74.94	130
Chicken_George	-11.11	61	Anthropic_Dreams	-74.94	131
BioShell	-11.68	62	Foldit	-74.94	132
Distill	-12.68	63	TAU_Course	-75.26	133
MeilerLab	-12.84	64	Wiskers	-75.32	134
wfCPUNK	-13.47	65	GoScience	-75.38	135
FLOUDAS_SERVER	-13.90	66	MICROGSIMU	-75.71	136
FLOUDAS_A3	-14.29	67	TAUbioinfounit	-77.24	137
rluethy	-14.91	68	MEAMT-group	-77.56	138
Cornell-Gdansk	-14.97	69	Nanoworld_Laboratory	-78.00	139
Bates_BMM	-15.11	70	MBBS	-78.00	140

Chapter 5

AIMOES: Archive information assisted multi-objective evolutionary strategy

5.1 Method

Evolutionary algorithm (EA) is the population-based optimization method, fitting the Darwinian principles of natural selection. Since the remarkable work of Schaffer [125], applying EAs to solve MOOPs have aroused a growing interest of researchers. The proposed AIMOES follows the common algorithmic framework applied in NSGA-II [126]: executing selection operators based on Pareto dominance and mutation operators to produce offspring iteratively, aiming to optimize the multi-objective function. The (1+2)-ES [127] is used to constitute the core of AIMOES. To enhance the performance of the proposed method, a global mutation operator is designed to reuse past search experience stored in the elitism archive. The archive is created to maintain the non-dominated solutions, and it is serviced as a knowledge database to provide information for evolution.

5.1.1 Main procedure

We use the (1+2)-ES because the process of evaluating a conformation of the protein is time-consuming to reduce the times of evaluation in a single iteration. The main procedure of AIMOES is shown in Fig. 5.1. In the initialization of solutions phase, a random conformation is created for each solution by setting every torsion angle with a random value satisfying the constraints shown in Table **??**. Initially, the archive A is empty and all the randomly generated solutions are added into A after the initialization phase.

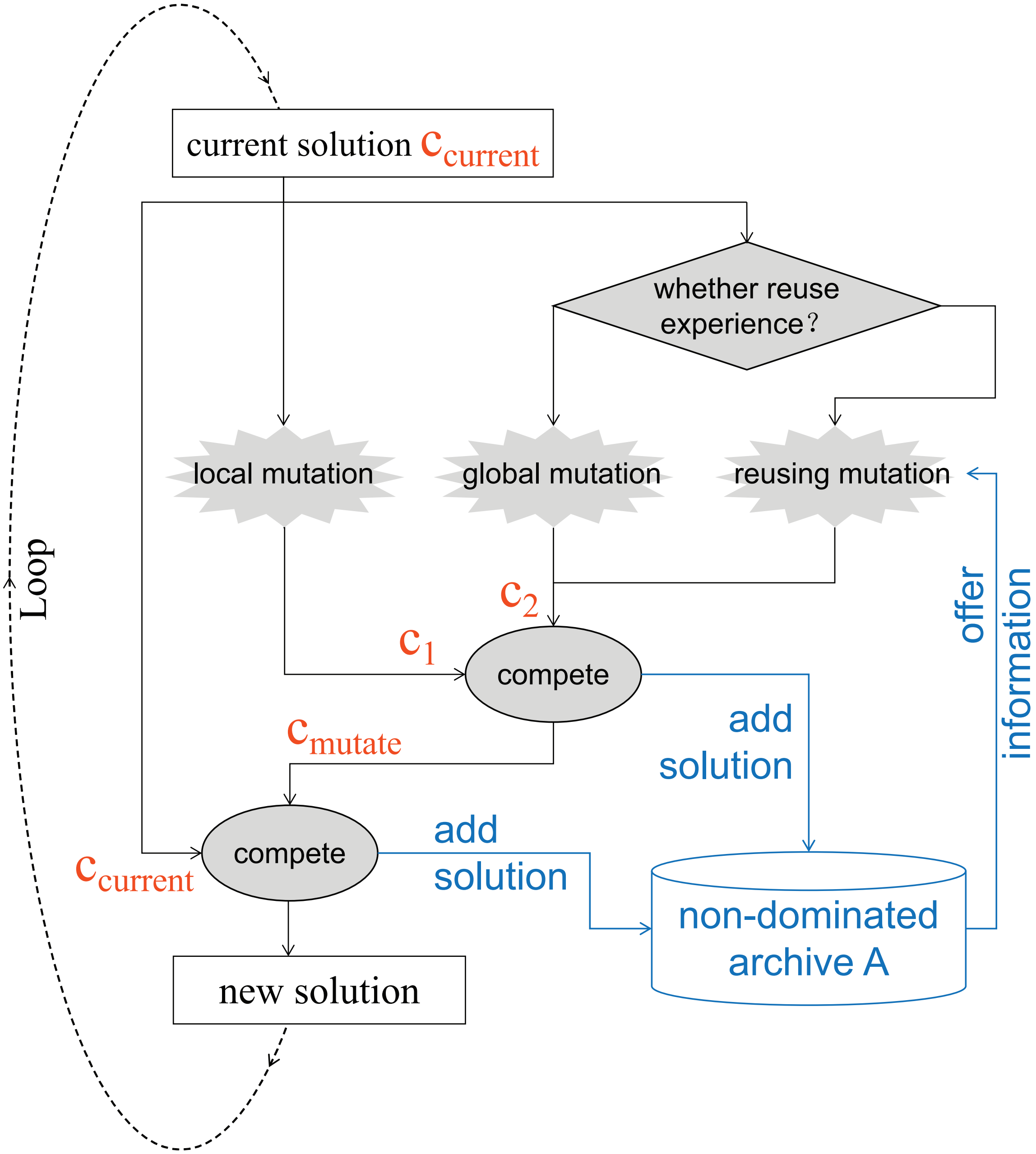


Figure 5.1: Main procedure of AIMOES.

Subsequently, the algorithm starts its evolution phase. First, two mutated solutions c_1 and c_2 are generated from the current solution via global and local mutation operators respectively. Then, they are evaluated by the multi-objective function, and compete for survival. The worse one is added to the archive A if it matches the joined condition described in Section 5.1.5, and the better one (now we call it c_{mutate}) competes with $c_{current}$ for survival.

For c_{mutate} and $c_{current}$, if one dominates the other, then the dominated one survives as the new current solution and the other will be discard. If neither dominates the other, we will refer to the information stored in A to determine which is better. If c_{mutate} dominates any one in the archive A, c_{mutate} will take place of $c_{current}$, because $c_{current}$ is non-dominated with any one in A and c_{mutate} should be considered to be better. At last, the algorithm terminates when the number of iteration is reached.

5.1.2 mutation operators

Three mutation operators are designed in the mutation phase shown in Fig. 5.1.

Local mutation The first one is local mutation, which changes the conformation of a protein slightly by perturbing its torsion angles with some randomly chosen residues. The number of chosen residues is:

$$N = 1 + (\frac{L}{4})e^{-\frac{t}{T_{max}}}, \tag{5.1}$$

where L is the length of the protein, t is the current iteration number, T_{max} is the maximum allowed number of iterations. It is clear that N decreases as t increases. The torsion angles of the protein are perturbed by:

$$\varphi_{new} = \varphi_{old} + \lambda\tau, \tag{5.2}$$

where φ_{new} and φ_{old} are the torsion angles of the protein before mutating and after mutating, respectively. λ is a scale factor and set to be 2.0 in the study. τ is a random number obeying the normal Gaussian distribution.

Global mutation Two types of global mutation are used in the work. The first one dramatically changes the current conformation by reseting all torsion angles with randomly chosen residues. The second one reuses the past search experience maintained in the archive A. At one point, only one global mutation operator is carried out. We set the probability of the first mutation operator to be implemented as:

$$M_1 = e^{-\frac{t}{4T_{max}}}. \tag{5.3}$$

It is clear that the probability of the first mutation operator decreases from 100% as the number of iterations t increases. Two reasons drive us to decrease the probability of the first mutation operator gradually. First, more use of the first mutation operator can enhances exploration capability of the proposed algorithm at the beginning. Second, there are small amounts of non-dominated solutions in the archive A at the beginning, thus little past search experience can be exacted by the second mutation operator from A.

5.1.3 Reusing search experience

Avoiding being trapped in local minimum search points is an important issue in conventional conformation search methods for solving PSP problem [128, 129, 130]. In order to overcome this shortcoming, many strategies have been proposed in the literature such as replica exchange [14], multi-canonical-ensemble [131] and Monte Carlo plus minimization strategy [64]. The common characteristic of these methods are that the hidden information among different individuals are effectively utilized. It is essential that designing an effective search strategy is necessary for improving the performance of an algorithm and useful information hidden in past search process should be incorporated into such design. As we can see, the first global mutation pays strengths to changing only one residue at one time. This simple mutation operator makes the current solution trapped in local optimum easily. An alternative way is to change many residues simultaneously and dramatically. But such a way would result in a low acceptance of the mutation operator. Considering that the topological structure of a protein can be determined by several backbone torsion angles in secondary structure coils coarsely, more attention should be paid to these angles, rather than all torsion angles of a residue.

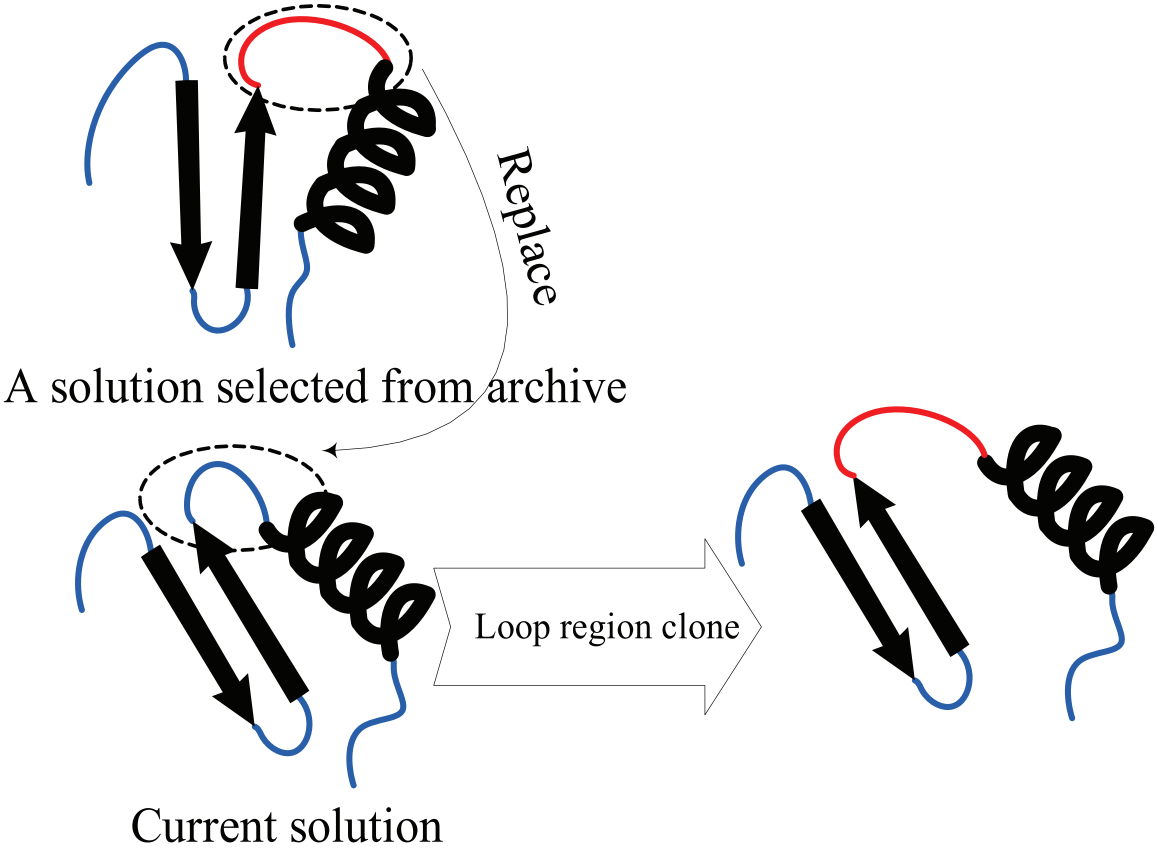


Figure 5.2: The global mutation operator reusing the past search experience.

In this study, a novel mutation operator which can effectively reuse the past search experience accumulated in the archive A is proposed. This reusing mutation operator is illustrated in Fig. 5.2. It takes the current solution and a "suitable" solution selected from A as input. A randomly selected loop region (the secondary structure of residues are coils) is selected, and the backbone torsion angles (ϕ, ψ) in the "suitable" solution are cloned to the current solution, generating a offspring. It is similar to the well-known two-point crossover operator [132], but there is only one offspring created. After mutating, the conformation changes in topology-level.

The method of selecting the "suitable" solution is explained as in the following. First, 30 solutions are randomly selected form A and the similarity between every pair of these solutions are calculated. Then the protein in the least crowded region is selected, which has the furthest average distance to all other selected solutions. This selection method is similar to the classic niching method, i.e. crowding [133]. The difference is that, a small set of samples are taken from the current population randomly in crowding method, but the comparison solutions are selected from the archive A in this study. In this way, these solutions in A in less crowded region are favored and some fragments of them are injected into the current solutions for further searching. As a result, more optimal solutions close to these solutions can be located and the population diversity of A is preserved.

5.1.4 Similarity between two proteins' conformations

How to evaluate the similarity between two objects is usually a crucial issue in the field of computational intelligence. Different metrics have been introduced, such as Hamming distance, Euclidean distance, squared Euclidean distance and cosine similarity. These metrics can reflect the different degree of two objects in genotype or phenotype space. In this study, a modified Euclidean distance is used in genotype space to calculate the similarity.

The backbone torsion angles of a protein conformation can be uniquely described as a series of torsion angles (ϕ and ψ):

$$V = \{x_1, x_2, x_3, x_4, ..., x_{2L}\}, \forall x \in V, -180^\circ \leq x \leq -180^\circ, \tag{5.4}$$

where L is length of protein. As indicated in [134], decision variables play different effects in the evolutionary progress, and different types of decision variables should be treated separately. For protein conformation, to reduce the complexity and describe simply, we bias the torsion angles of a residue in the secondary structure coil. The topology structure of a protein is almost determined by these residues. a subset U of V is created by picking out the backbone torsion angles in secondary structure coil, as following:

$$U = \{y_1, y_2, y_3, y_4, ..., y_m\}. \tag{5.5}$$

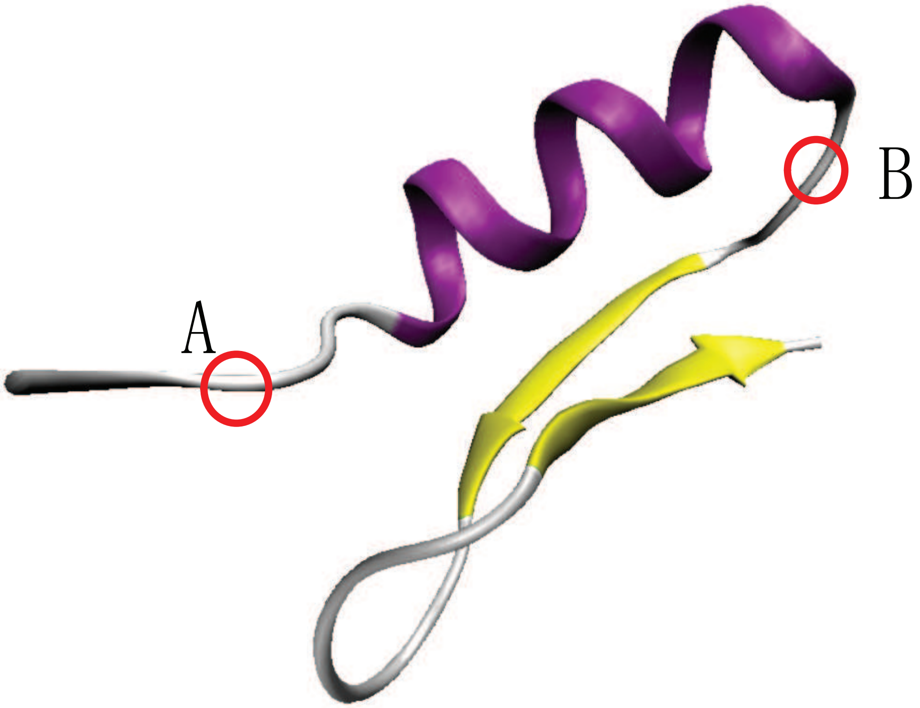


Figure 5.3: The backbone torsion angles in different region play different contributions to determine the topology structure. For example, The angles in region B are more crucial than the ones in A.

As a note, the elements (torsion angles) in V and U are arranged in order by their position in sequence. m is the number of ϕ and ψ in coil.

The similarity of two protein conformation can be coarsely evaluated by calculating the modified Euclidean distance of U. Considering the residues at both ends and the ones in middle, they are unbalanced for playing different contributions to determining the topology structure shown in Fig. 5.3. The one in middle are more crucial, and should be increased weighting coefficient for calculating. Thus, the modified distance in decision space is as following:

$$D = \sqrt{\sum \eta_i {d_i}^2}, \tag{5.6}$$

where i the order number of angle in U. d_i is the distance between angle i of two matched confrontation. The weighting coefficient η_i is a position-dependent factor,

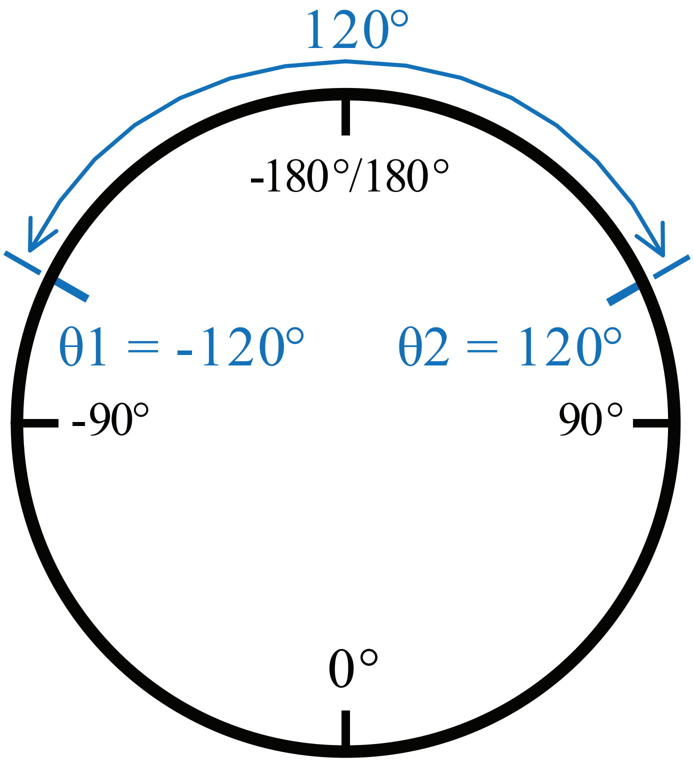


Figure 5.4: A example of calculating the distance between two angles. Because of the circle structure, the distance value between θ_1 and θ_2 is 120°, rather than 240°.

proportional to a normal distribution:

$$\eta_i = e^{\frac{-(i-\mu)^2}{2\delta^2}}, \mu = \frac{|U|}{2}, \delta = \frac{|U|}{4}. \tag{5.7}$$

Every element in U is an angle, in the range $[-180°, 180°]$. Calculating the difference between two angle is different from the Euclidean distance, because of the circle structure. In a circle, the inferior arc of two angles is defined as the distance between two angles, always smaller than 180°. Fig. 5.4 shows the distance between $\theta_1 = -120°$ and $\theta_2 = 120°$ is 120°, rather than 240°. the instance between two angles is calculated as:

$$d(a,b) = \begin{cases} |a-b|, & |a-b| \leq 180° \\ 360° - |a-b|, & |a-b| > 180°, \end{cases} \tag{5.8}$$

where a, b are torsion angles and $a, b \in [-180°, 180°]$.

5.1.5 Non-dominated solutions in the archive A

It is a common practice to incorporate external elitism populations in classical MOEAs [110, 135]. The external population retains all non-dominated solutions produced along evolutionary process. An archive A is created to maintain a set of non-dominated solutions along evolutionary process in the work. At each iteration, a solution c^* can be added to archive if (1) A is not full or c^* is not dominated by any one in it, (2) c^* dominates some solutions in A and these dominated solutions are removed from A, and (3) A is full and c^* is non-dominated by any one in A. Then, 30 solutions are randomly selected, and the one with the smallest average distance to all other selected solutions, thought in most crowded region, is replaced by c^*.

The archive A serves three purposes in the evolution process. Firstly, it stores and updates all non-dominated solutions along evolution process. Secondly, it can assist the selection operator to determine which is better between two solutions. Thirdly, It can retain structured knowledge buried in the non-dominated solutions, which is transfered from past search experience. The mutation operator can reuse fragments captured from these non-dominated solutions stored in A. At this moment, it can be seen as fragment library, similar to fragment assembly strategy used in Rosetta [64]. Different from Rosetta, these fragments are obtained from past optimal solutions, rather than known proteins.

5.1.6 Decision maker

Similar to typical FMs, the proposed method outputs a large number of solutions stored in the archive A. The method how to select representative structures from a set of decoy structures is also worth discussing [136]. In order to maintain integrity, a method based on hierarchical clustering is proposed. It is carried out as follows:

- calculate the distance of every pair of structures in A. Usually a dissimilarity matrix is created.

- use complete linkage clustering (furthest distance) and group these structures into a hierarchical cluster tree.

- cut the hierarchical tree into clusters at a given cutoff value.
- pick out the centroid of the clusters with largest size.

We try to use two types of distance metrics to generate the dissimilarity matrix. The former is the distance of two structures described above in genotype space. The cutoff value is set to 100 empirically. The later is the RMSD$_{C_\alpha}$, described in Section 5.2. It can be seen as the distance of two structures in phenotype space. The cutoff value is set to 8.

5.2 Experiment and discussion

5.2.1 Target proteins

Table 5.1: Information of target proteins.

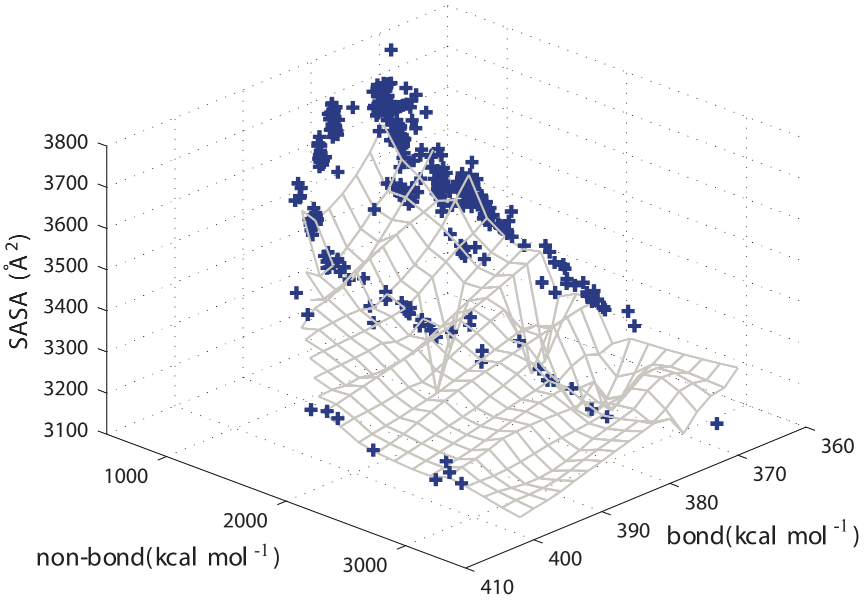
PDB ID	SS*	Length	PDB ID	SS*	Length	PDB ID	SS*	Length
1AB1	α/β	46	1I6C	β	39	2JUC	α	59
1AIL	α	73	1IGD	α/β	61	2MR9	α	44
1BDD	α	60	1K36	β	46	2P5K	α/β	64
1DFN	β	30	1MSI	β	70	2P6J	α	52
1E0G	α/β	48	1Q2K	α/β	31	2P81	α	44
1E0M	β	37	1SXD	α	91	2PMR	α	87
1ENH	α	54	1ZDD	α	34	3V1A	α	48
1F7M	β	46	2GB1	α/β	56			
1G26	β	31	2JZQ	α	57			

* SS: secondary structure classification

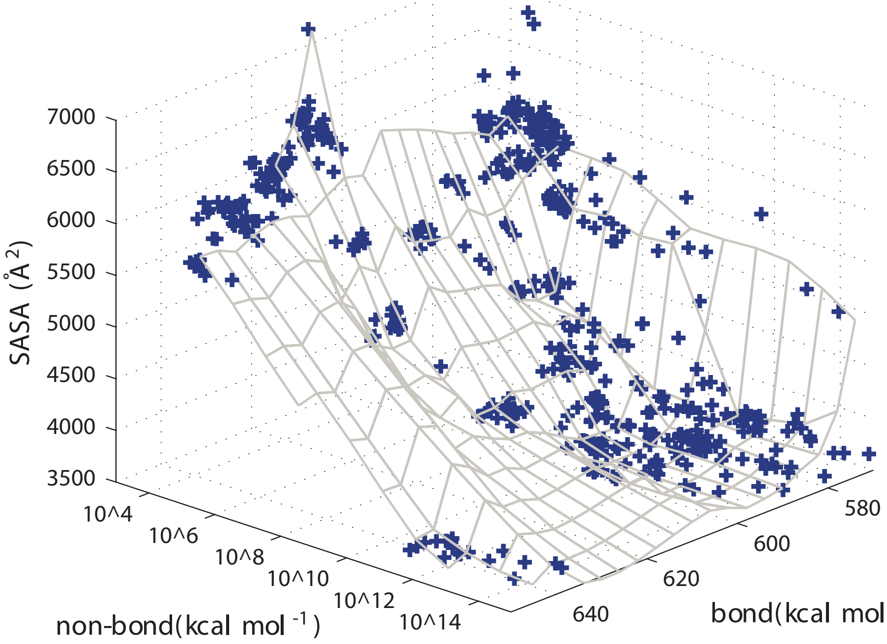
Table 5.1 lists the set of 25 tested proteins. Their lengths vary from 30 to 91. The structural classes of the test proteins contains α, β, and α/β [137].

5.2.2 Experimental environment

We run the program on the Linux 64-bit system with four 3.40GHz Intel Core(TM) i5 processor and 8GB memory. For each target protein, we set the maximum number of iterations 40000. Typically, this program takes about 20 hours for a single run.



(a) 1AB1

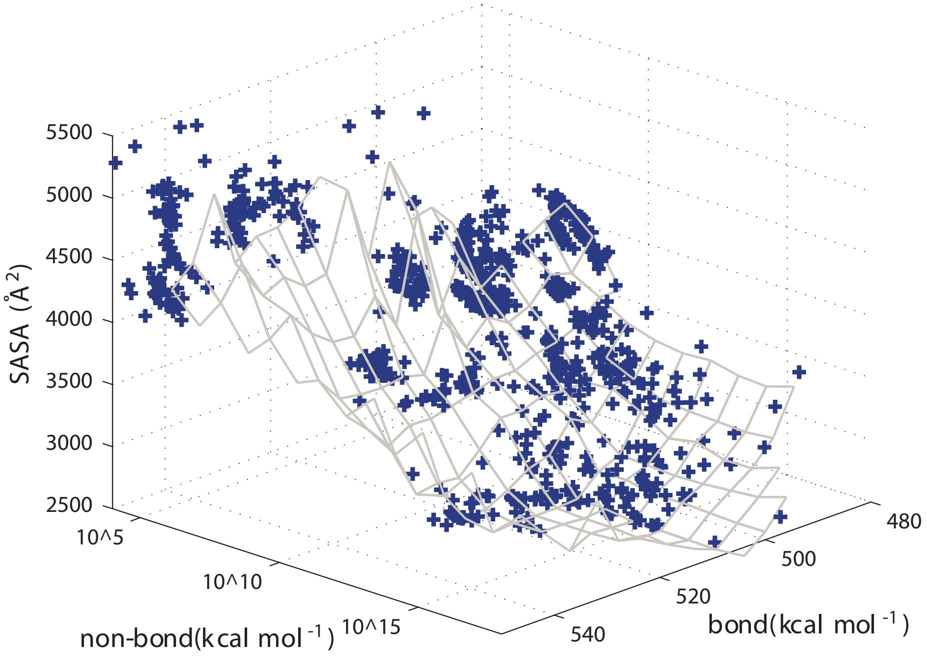


(b) 1AIL

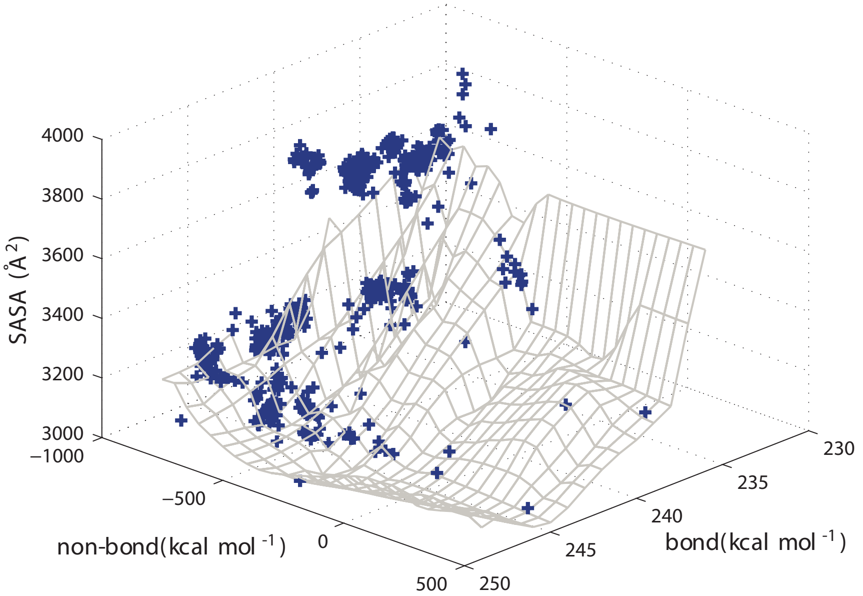
Figure 5.5: Pareto fronts for target proteins (1).

5.2.3 Predicted results

We run the proposed algorithm on these proteins to predict their structures. For every protein, an archive maintained non-dominated solutions is generated at last.



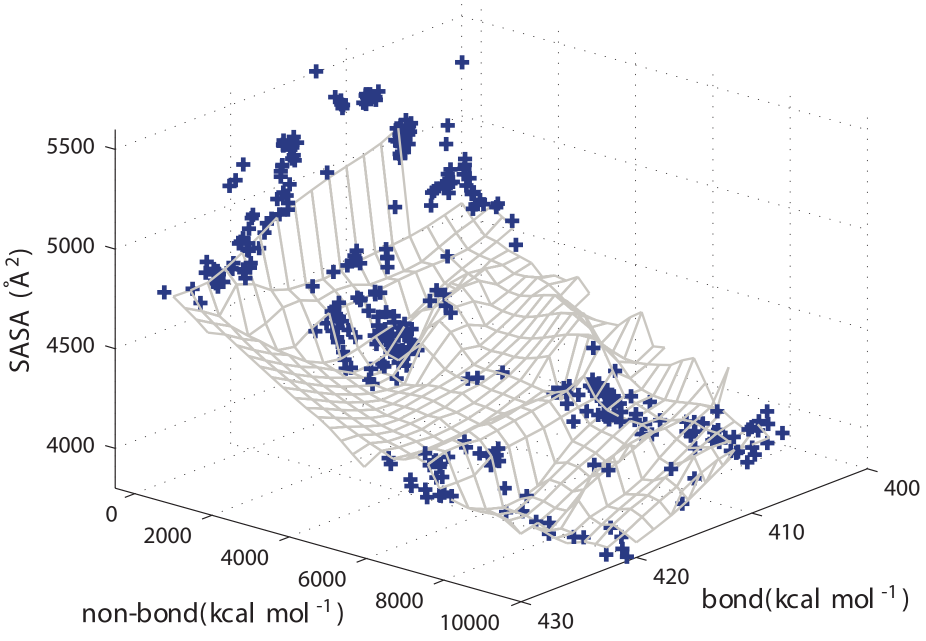
(a) 1BDD



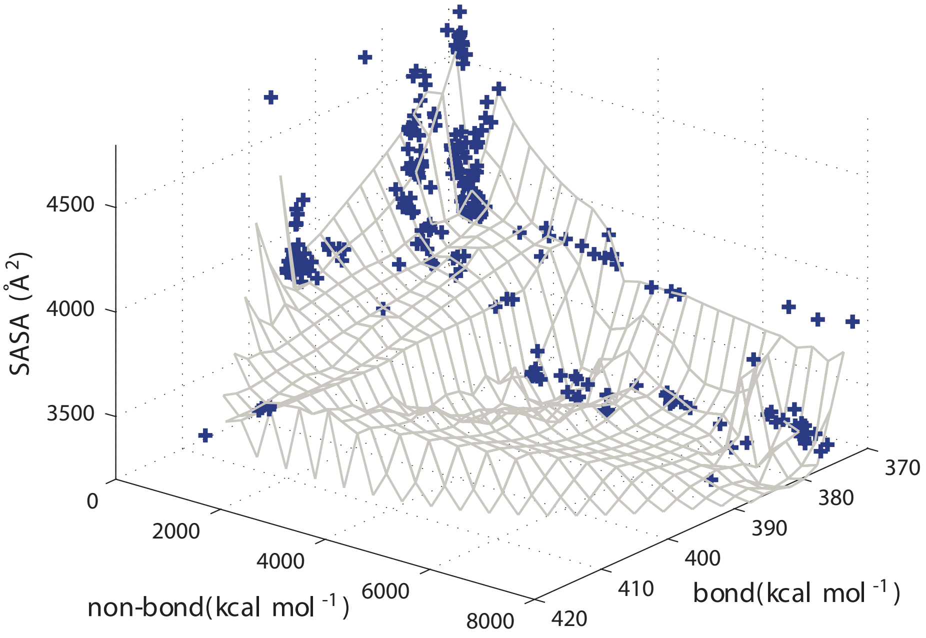
(b) 1DFN

Figure 5.6: Pareto fronts for target proteins (2).

The Pareto fronts of each protein are shown in Figs. 5.7 ∼ 5.16, which can exhibit how approximate Pareto fronts are obtained. It is noted that, the gray surfaces in



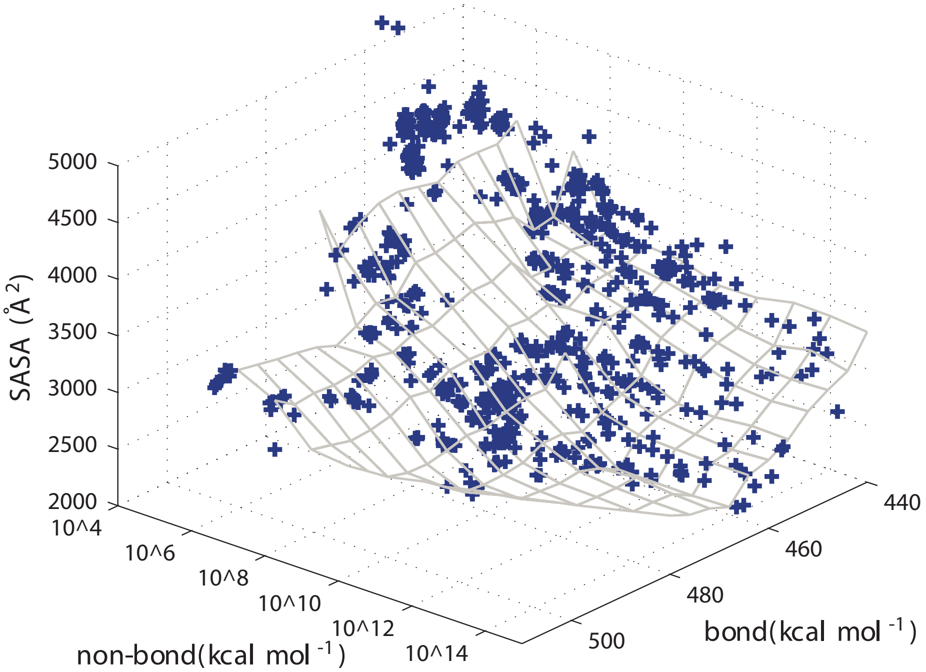
(a) 1E0G



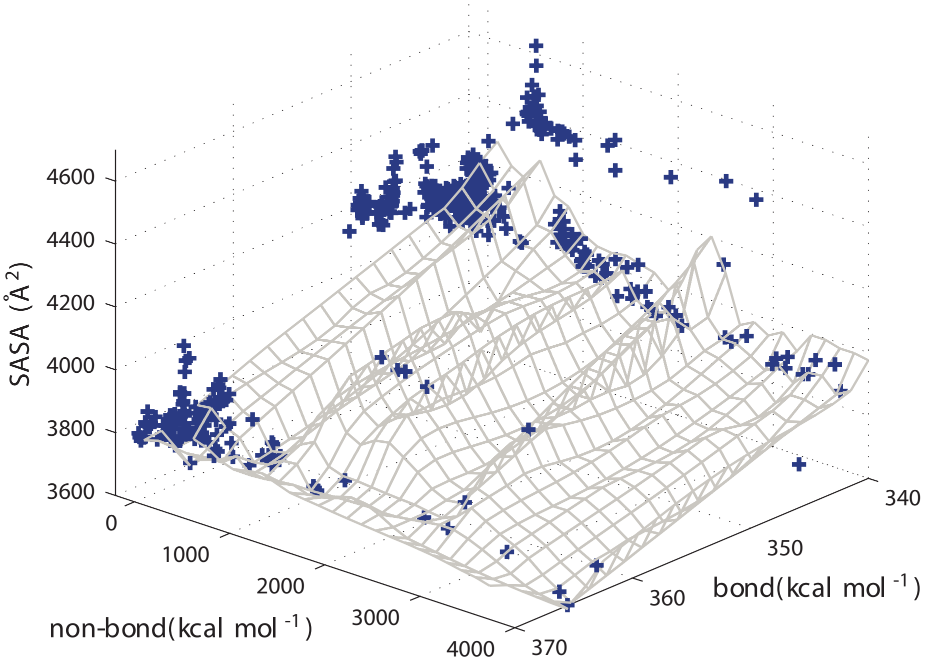
(b) 1E0M

Figure 5.7: Pareto fronts for target proteins (3).

these figures are not the real surfaces of Pareto fronts, they assist us easily viewing the three-dimension Pareto fronts. From these Pareto fronts, we can see that non-bond energy has a bigger change range than bond energy. Specially, For 1AIL, 1ENH, 1MSI,



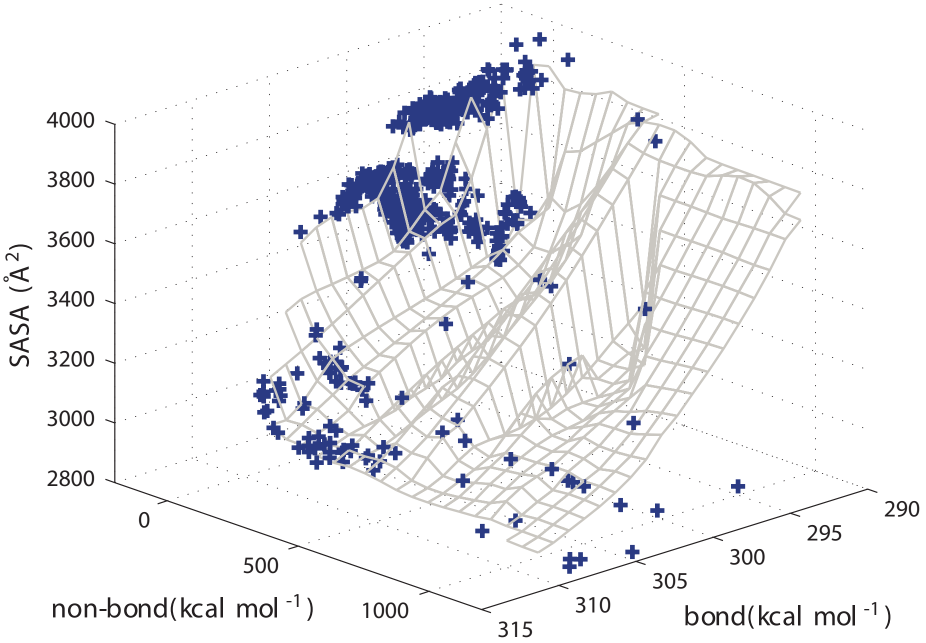
(a) 1ENH



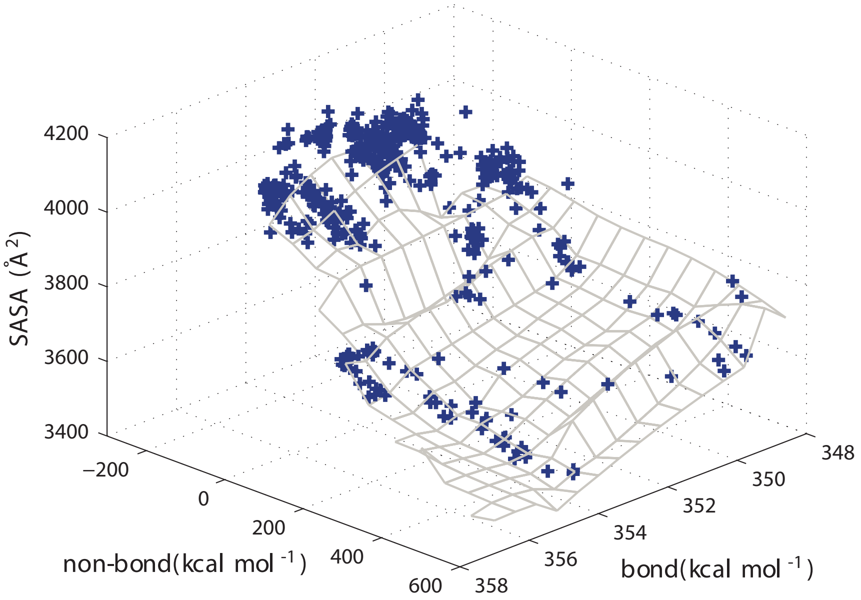
(b) 1F7M

Figure 5.8: Pareto fronts for target proteins (4).

2JUC and 2PMR protein, non-bond energy term changes sharply at large scales. It proofs the necessity of decomposing CHARMM22 into bond energy and non-bond energy. Because large variation range of non-bond energy can hide the change of



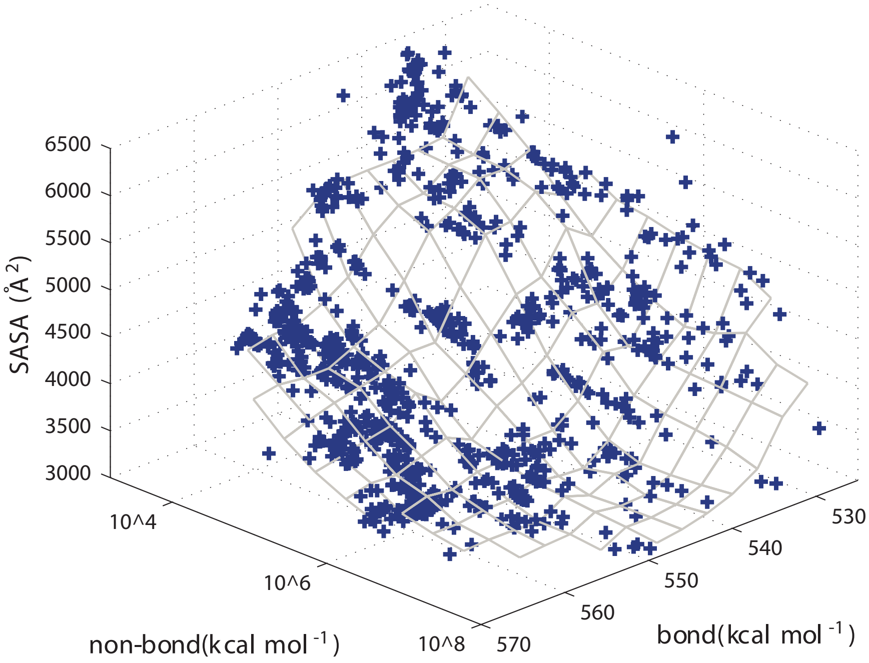
(a) 1G26



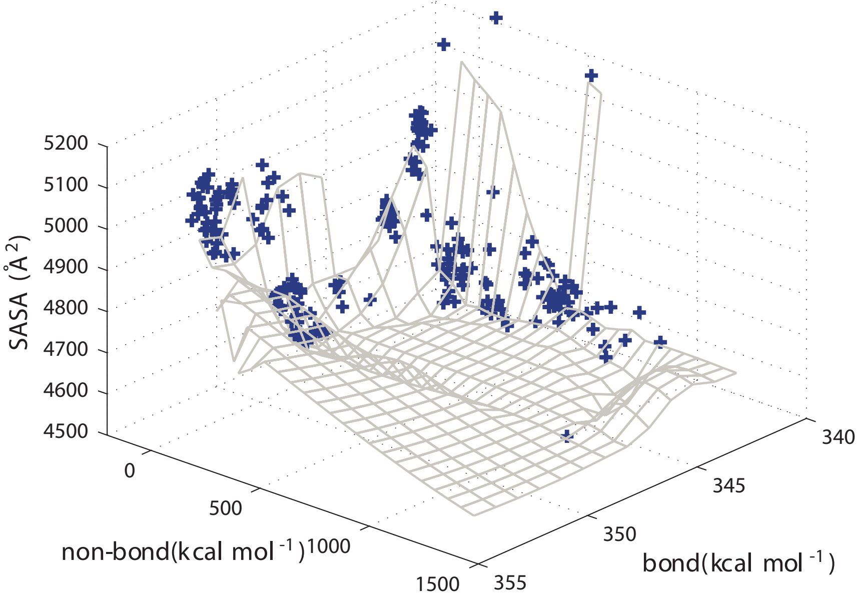
(b) 1I6C

Figure 5.9: Pareto fronts for target proteins (5).

bond energy. Moreover, the rugged Pareto fronts surface indicated the complexity of designed protein energy function and the complexity of PSP. Modeling PSP problem as a multi-objective optimization problem is also hard to solve comparing with single-



(a) 1IGD

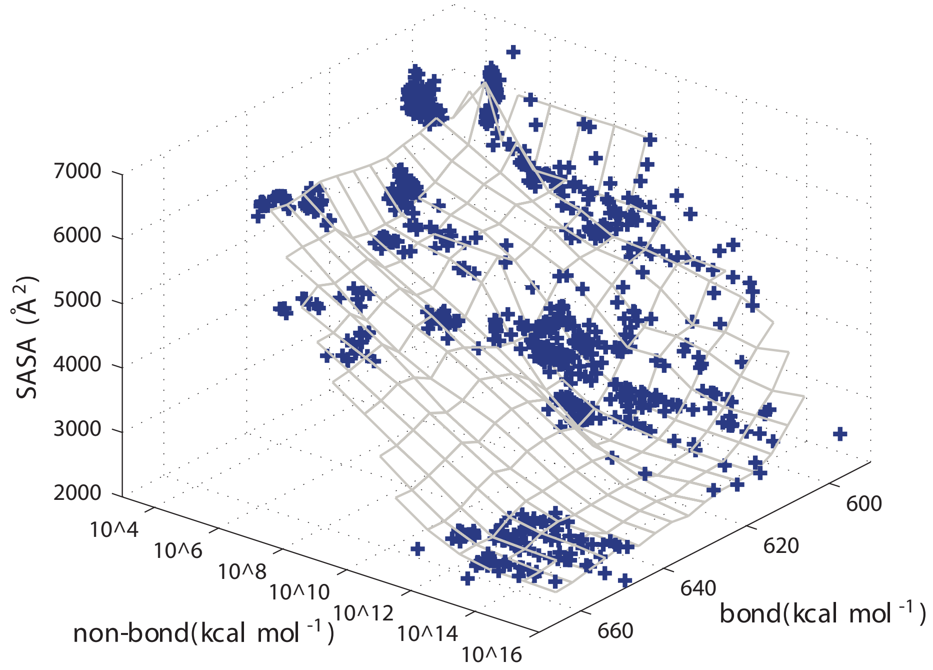


(b) 1K36

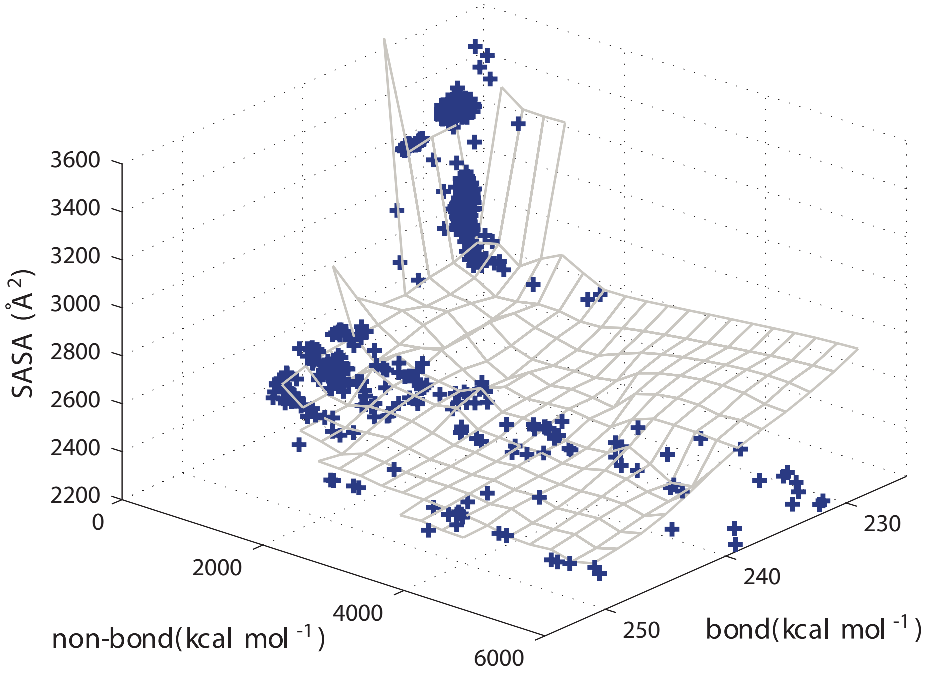
Figure 5.10: Pareto fronts for target proteins (6).

objective optimization.

We use the hierarchical clustering described above to cluster these solutions stored



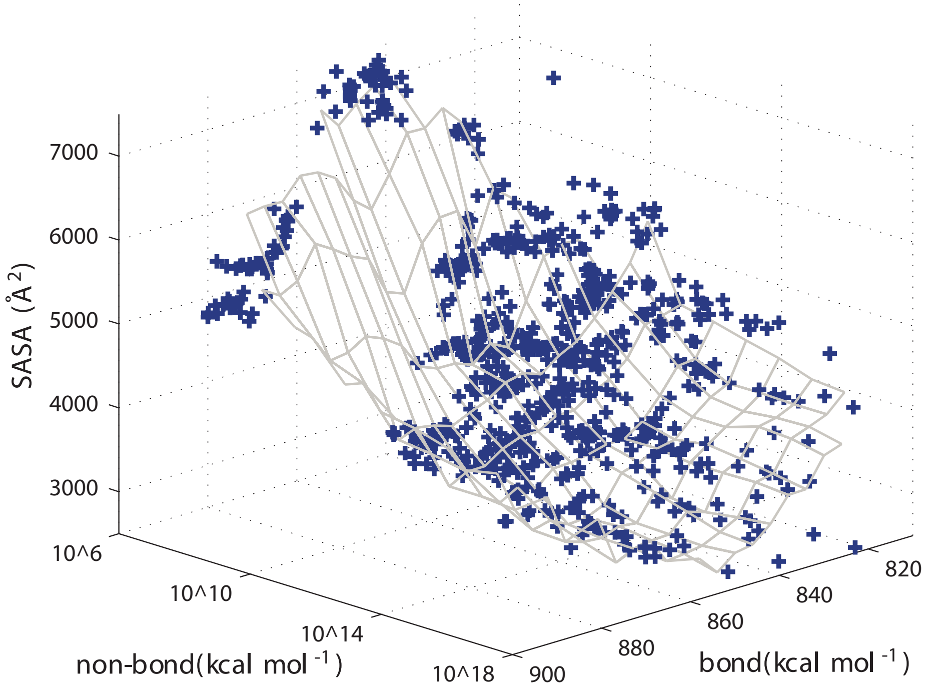
(a) 1MSI



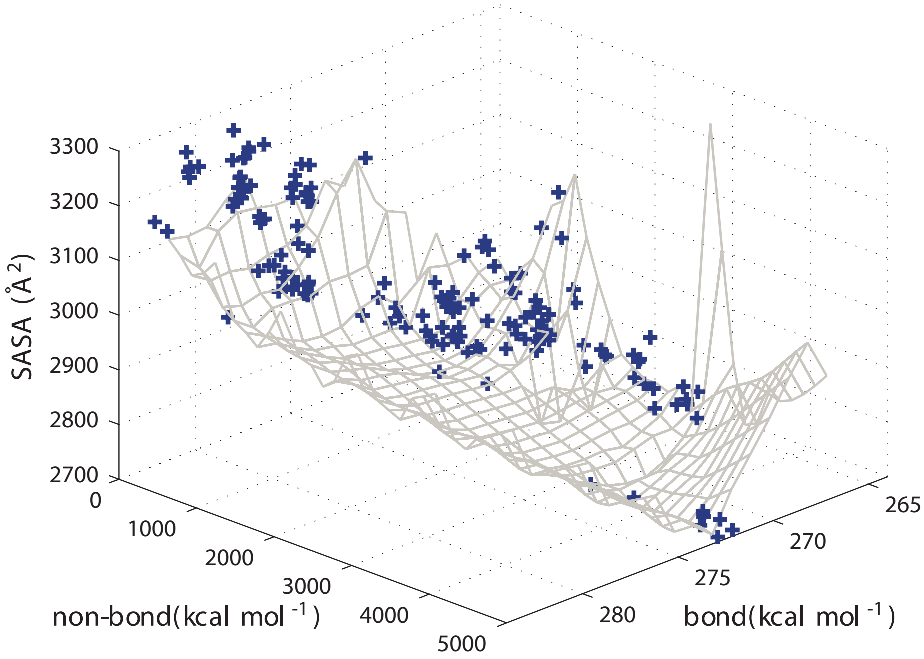
(b) 1Q2K

Figure 5.11: Pareto fronts for target proteins (7).

in the archive A in the genotype space and phenotype space, respectively. Table 5.2 reports the solutions with best RMSD$_{C_\alpha}$, cluster centroid with maximum size in genotype space phenotype space. We compare the results clustered in genotype and



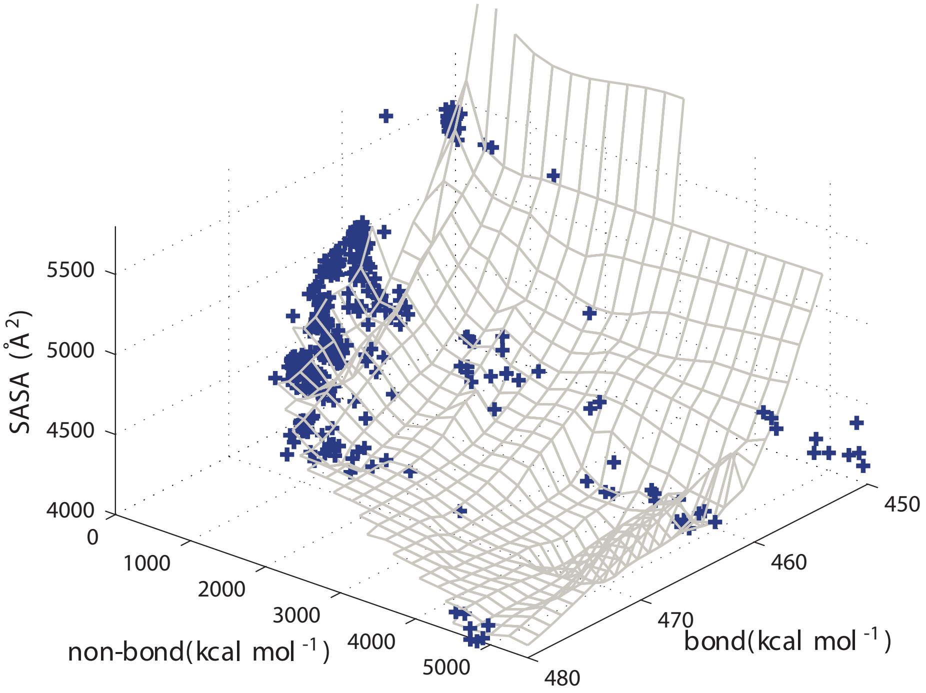
(a) 1SXD



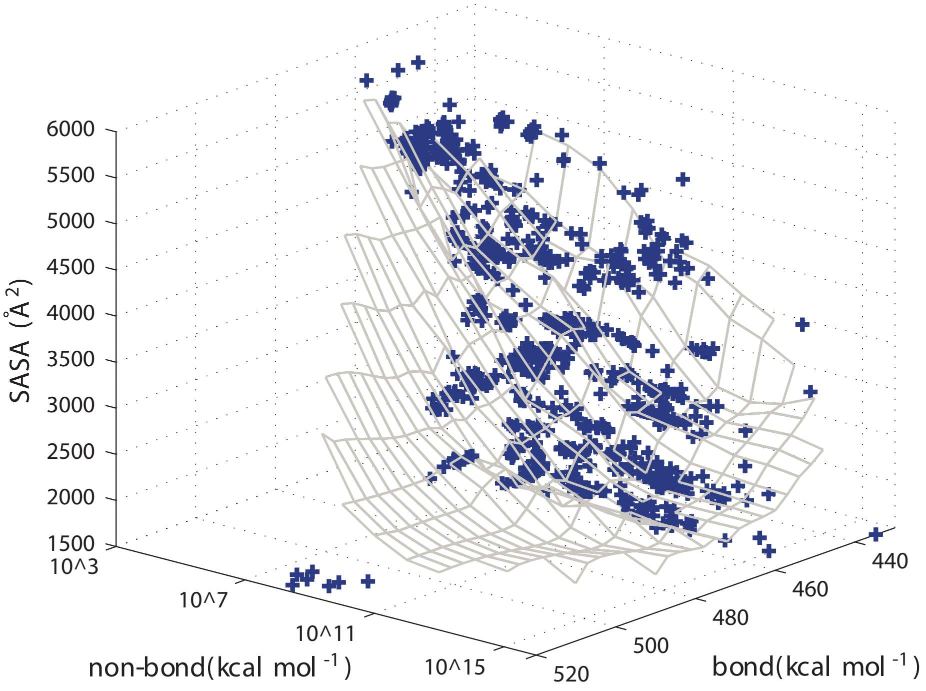
(b) 1ZDD

Figure 5.12: Pareto fronts for target proteins (8).

phenotype space respectively. It is interesting that the results clustered in genotype space are mainly better than in phenotype space. It is probably due to that the



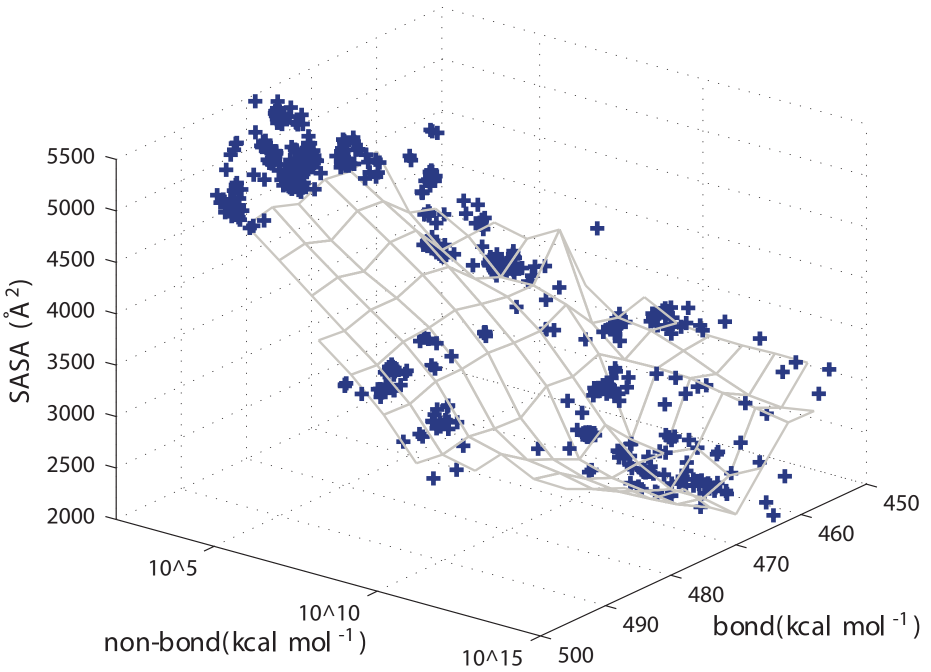
(a) 2GB1



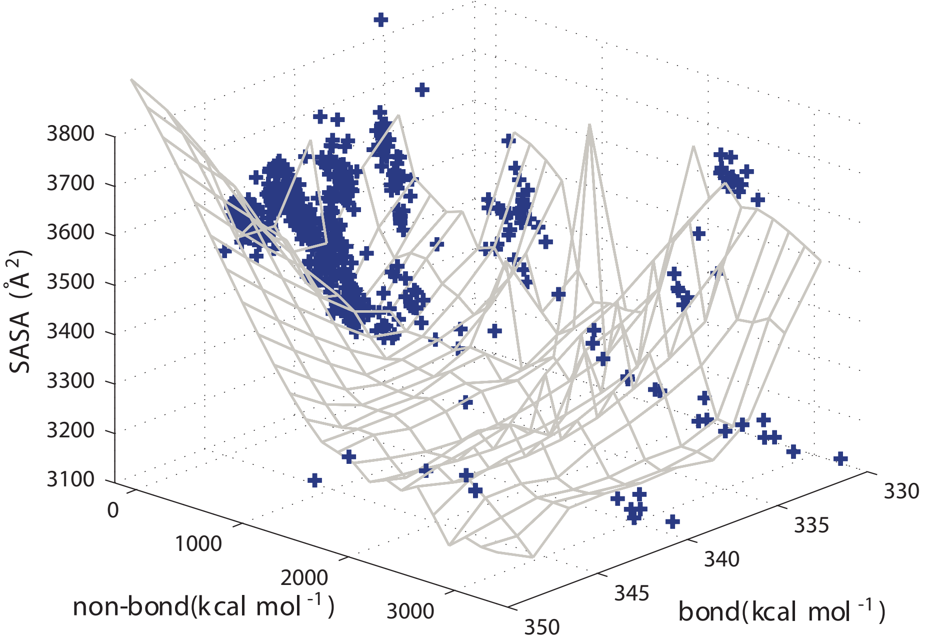
(b) 2JZQ

Figure 5.13: Pareto fronts for target proteins (9).

distance in genotype between two solutions reflects the essential difference of them. As a sequence, constructing a more pure hierarchical tree. Moreover, Fig. 5.18 shows



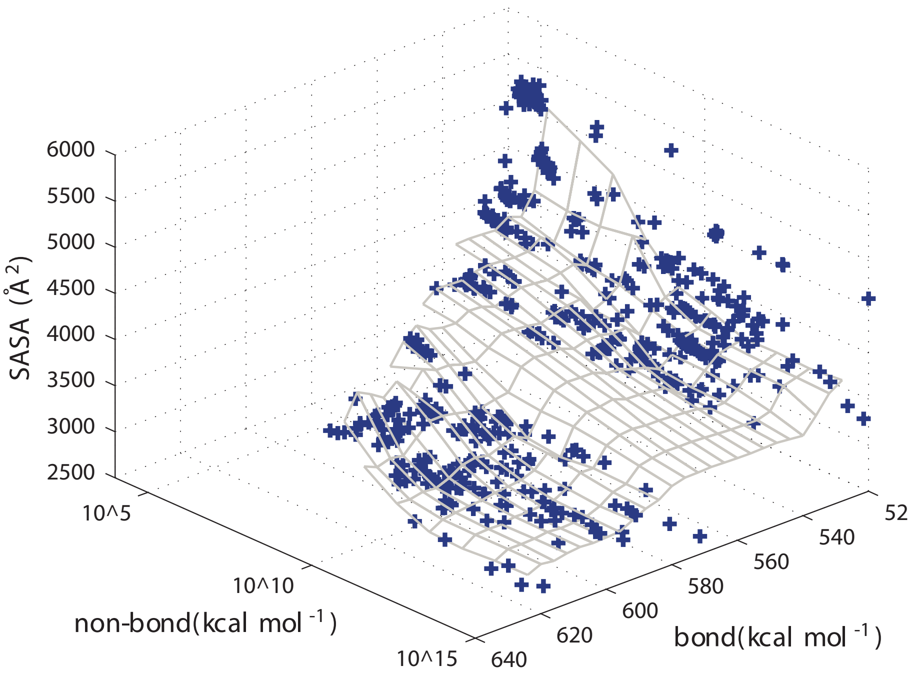
(a) 2JUC



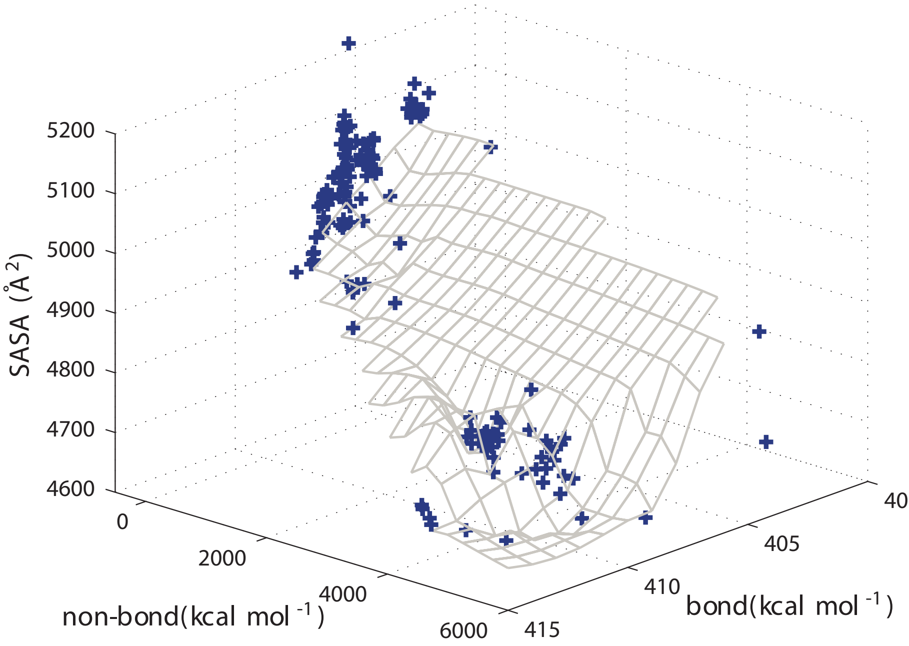
(b) 2MR9

Figure 5.14: Pareto fronts for target proteins (10).

the superposition of native structures and predicted structures (hierarchical clustered in genotype space). It suggests the power of proposed method to solve PSP since excellent or acceptable solutions can be obtained at last.



(a) 2P5K

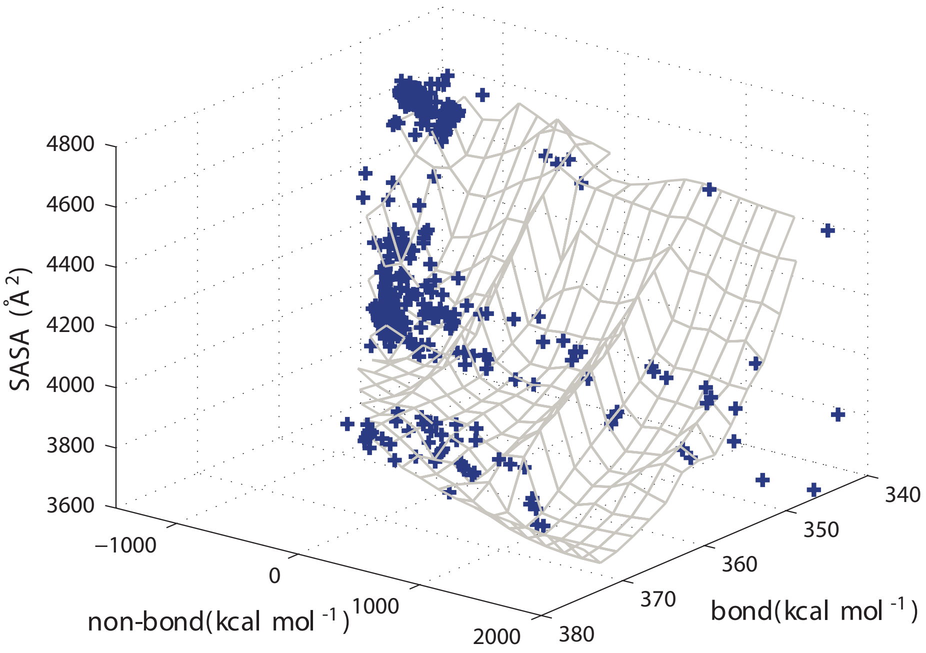


(b) 2P6J

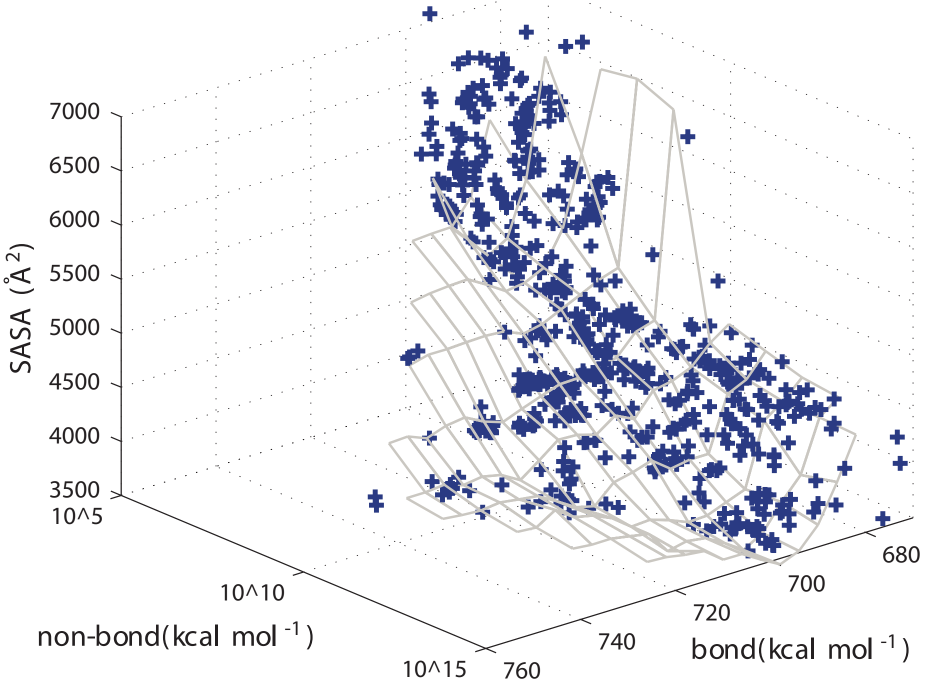
Figure 5.15: Pareto fronts for target proteins (11).

5.2.4 Comparing success rate of mutation operators

Designing effective conformation search is an critical issue in FMs. Besides, it is also important to design mutation operator with high acceptance in EA for improving



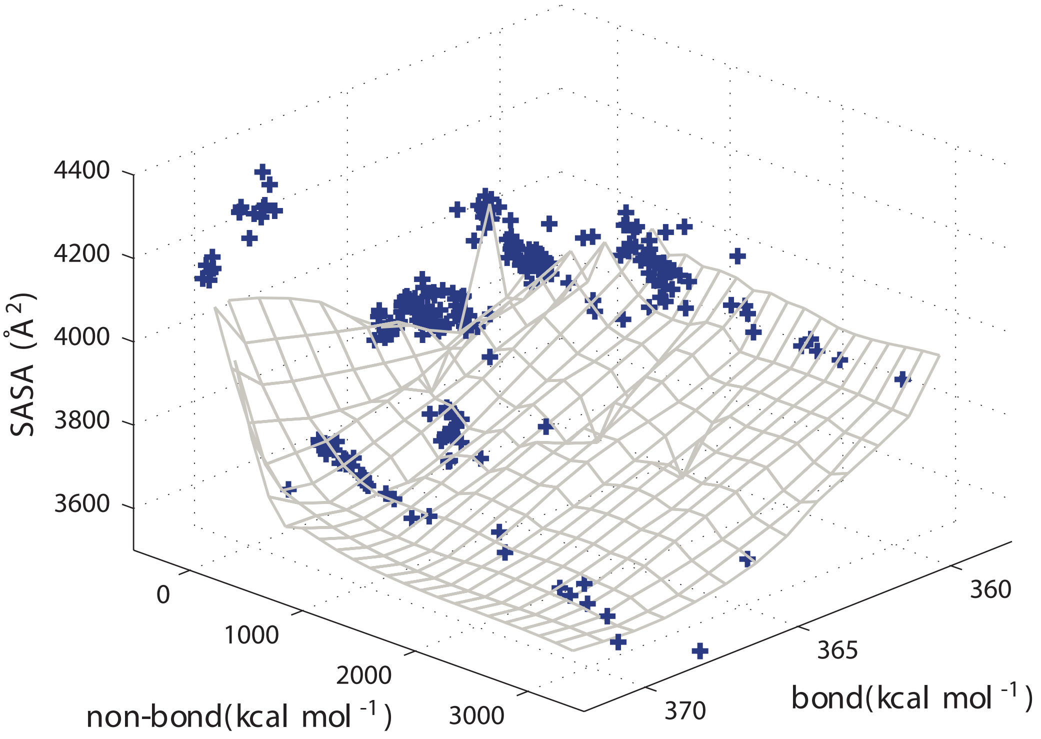
(a) 2P81



(b) 2PMR

Figure 5.16: Pareto fronts for target proteins (12).

efficiency. We compare the performance of three mutation operators. As we can see, two offspring are generated by local and global mutation and compete for survival.



(a) 3V1A

Figure 5.17: Pareto fronts for target proteins (13).

One will dominate the other, or neither dominates the other. We count the acceptance rate of a mutation operator, defined as the rate of one offspring dominates the other.

Fig. 5.19 shows that, local mutation operator is more likely to generate a better offspring than global mutation at 10%∼25%. Because it changes the current solution in a smaller magnitude. Comparing two types of global mutation operator, the reused operator have higher acceptance rate, even though it change more torsion angles in topology-level at a time. As the reused operator injects fragments of a non-dominated solutions stored in the archive A into current solution. These fragments existing in A can be seen as a form of past search experience. Learning from them can increase acceptance rate of the reused operator.

5.2.5 Comparative test

A comparative test was carried out to investigate how the reused strategy influences the predict result. We modified the proposed algorithm by removing the reused mu-

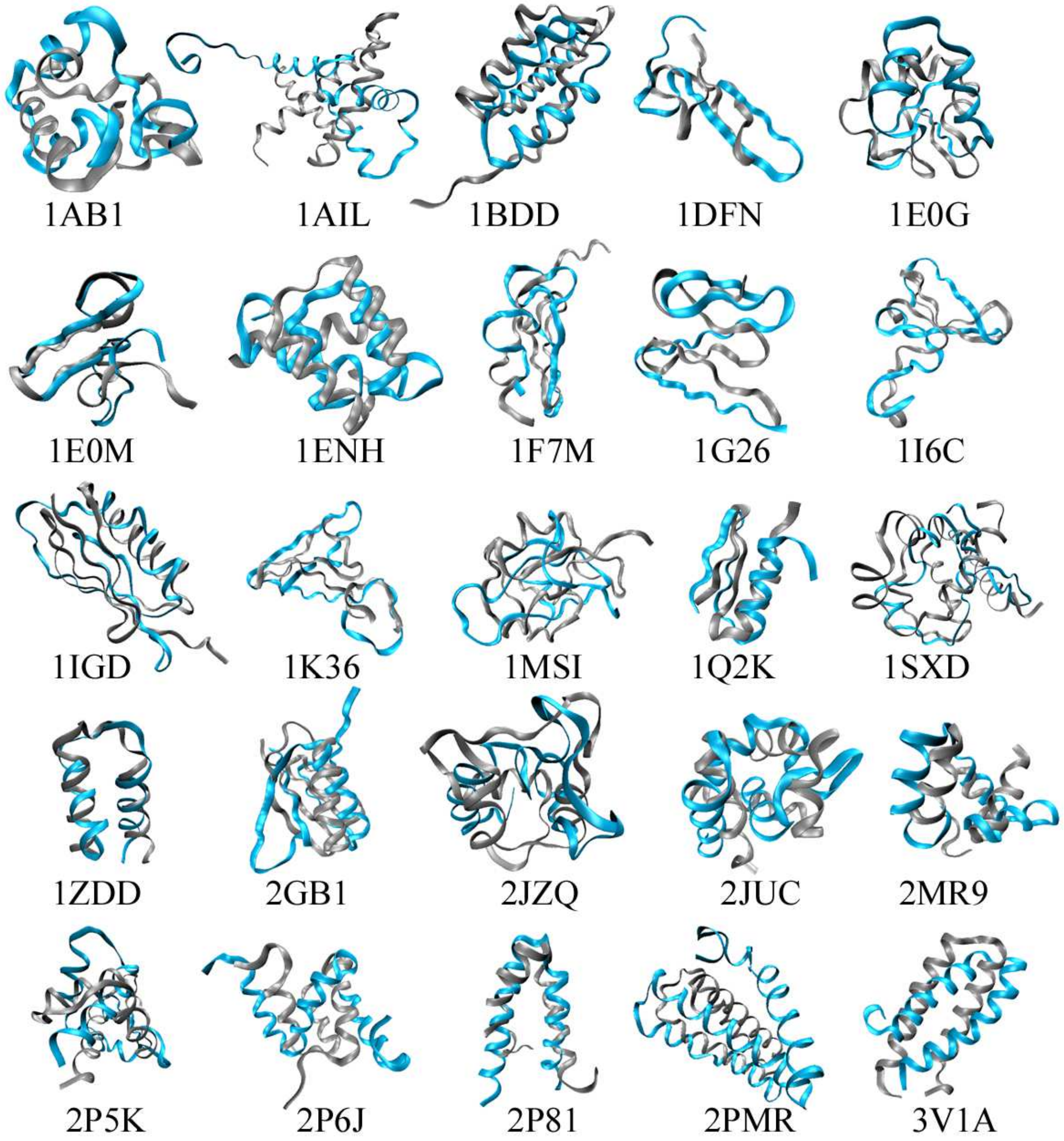


Figure 5.18: Superposition of the native and predicted structures which are the cluster centroids with maximum size in genotype space.

tation operator, and run it on these proteins again. The variations of RMSD_{C_α} value of decoy structures obtained by two strategy are showed in box-plot Fig. 5.20. As can be seen from Fig. 5.20, the incorporation of reused strategy allows the proposed algorithm to achieve more excellent solutions (lowest RMSD_{C_α}) than the one without reused strategy. For example, the proposed algorithm outperforms in 21 proteins of 25. Moreover, comparing the best quarter of RMSD_{C_α}, It is clear that the pro-

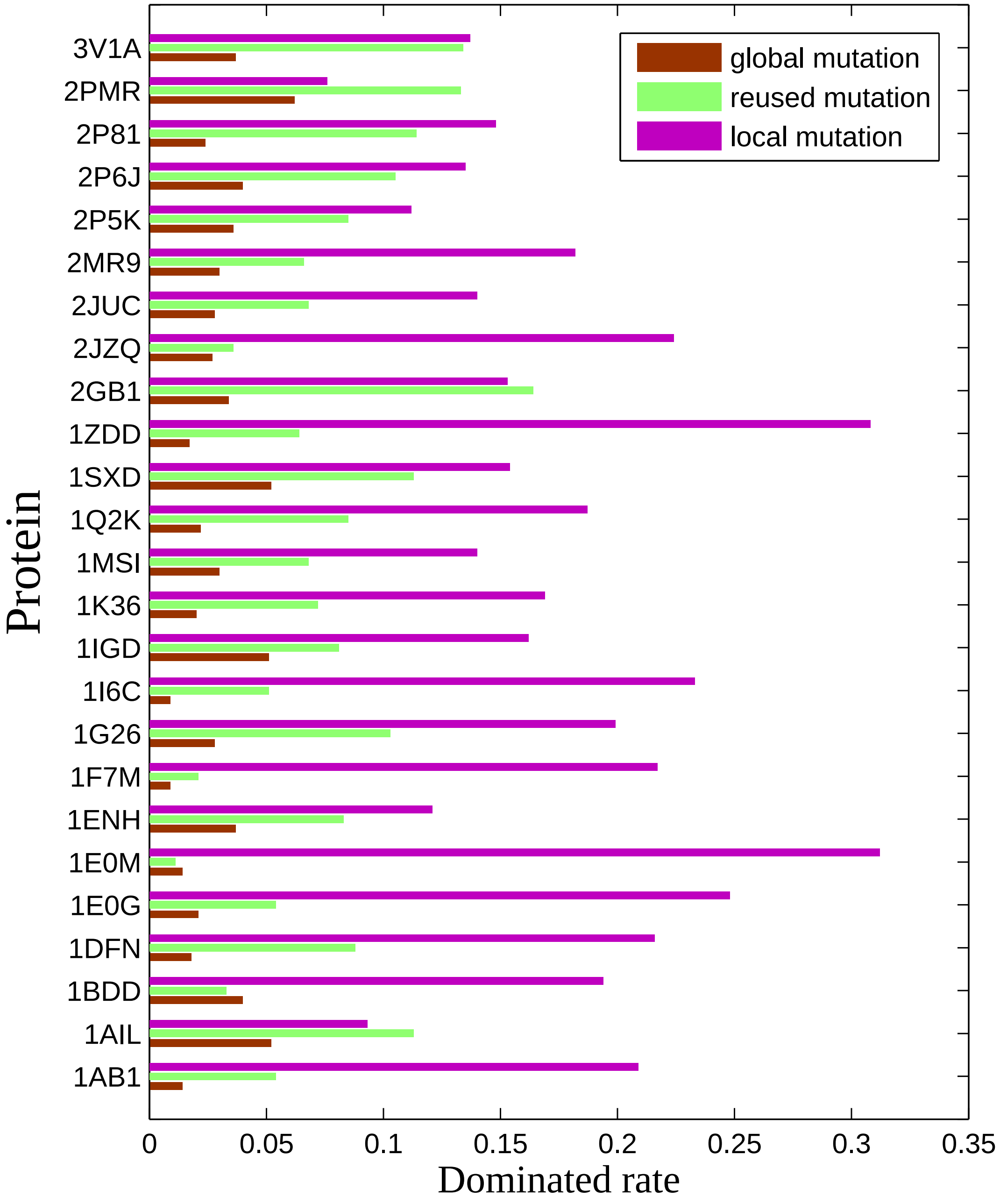


Figure 5.19: Dominated rate of three mutation operators, corresponding to Fig 5.1.

posed algorithm can reach and preserve more native-like structures in most case. The distributions of RMSD_{C_α} also exhibits the diversity of structures in the archive A, illustrating reused strategy can maintain diverse solutions with high accuracy. This is due to including niching method implicitly in selection phase, even though in genotype space. In short, the reused strategy can enhance the robustness end effectiveness of proposed algorithm to solve PSP.

5.2.6 Comparing with other works

We also compared the proposed method with other works in the literature. Specially, four evolutionary computation methods are compared. (1) I-PAES [25] is a modified version of PAES [110] to perform PSP. It decomposes CHARMM27 into two objectives and uses the MOEA to search for Pareto-optimal sets of conformations. (2) MEAMT is a MOEA, working with more than one subpopulation in parallel though tables. It is called multi-objective evolutionary algorithm with many tables [26]. Specially, it deals with four objectives through combination thereof, rather than Pareto dominance. (3) GA-APL [71] is an angle probability knowledge-based methods based on general genetic algorithms for PSP. Rosetta energy function [139] is used as the single-objective function to minimize. (4) Venske et al. applied ADEMO/D (an adaptive differential evolution for multi-objective optimization based on decomposition) to performing PSP [138]. CHARMM27 is decomposed into bond energy and non-bond energy as objective function.

Because there are not a standard test protein set, a strict comparison is impossible and empirical comparisons are executed. In addition, some methods lose the process of decision making. They only reported the the predicted structure with lowest RMSD$_{C_\alpha}$ in a set of decoy structures. A qualitative comparison is carried out in Fig. 5.21. Best RMSD$_{C_\alpha}$ or RMSD$_{C_\alpha}$ after decision making of each method according to protein length are drawn pot. The test proteins of each method with length smaller than 30 are excluded. The results after decision making are marked with DM. Linear regression lines between protein length and RMSD$_{C_\alpha}$ for each method are also plotted to view the relationship between RMSD and protein length. Three is no doubt that a longer protein corresponds to more hard prediction. According to the tendency of linear regression, the proposed methods exhibit the ability to produce better or competitive performance comparing with other methods.

Moreover, to make a quantitative comparison, a normalized root-mean-square distance RMSD_{100} [140] is introduced as follows:

$$RMSD_{100} = \frac{RMSD}{1 + \ln\sqrt{\frac{N}{100}}}, \tag{5.9}$$

where N is the length of two compared structures. RMSD_{100} can be seen as the degree of similarity of two structures when the length is normalized to 100. It can be used as the metric to compare RMSD values for proteins with different length.

Performance results are reported in Table 5.3. The number of test proteins are the reported proteins in each method (length ≥ 30). The length range of test proteins and the structural classes of the test proteins are reported in column 3 and 4 respectively. The evaluation times and average RMSD_{100} are reported in column 5 and 6. From Table 5.3, we can see that the proposed algorithm can produce more excellent solutions than I-PAES, MEAMT and GA-APL before decision making, even though with smaller evaluation times. After decision making, our proposed method also can provide competitive performance with ADEMO/D, considering ADEMO/D is lack of comprehensive test.

Table 5.2: Summary of RMSD$_{C_\alpha}$ of predicted results.

	MO3	AIMOES		
PDB ID	RMSD best[1] (Å)	RMSD best[1] (Å)	RMSD genotype[2] (Å)	RMSD phenotype[3] (Å)
1AB1	6.67	6.23	6.77	6.77
1AIL	7.96	6.69	9.63	9.97
1BDD	5.95	6.28	6.95	6.9
1DFN	4.89	4.68	7.65	7.68
1E0G	6.62	5.68	7.28	7.45
1E0M	5.72	5.73	5.94	6.5
1ENH	6.52	5.75	6.67	8.44
1F7M	7.2	7.91	9.71	9.71
1G26	5	4.49	5.57	6.49
1I6C	6.49	5.63	8.02	7.97
1IGD	7.62	6.95	7.19	7.98
1K36	9.07	7.07	10.15	10.12
1MSI	8.41	7.51	9.59	9.45
1Q2K	3.52	3.08	4.27	8.23
1SXD	10.22	8.34	12.12	11.19
1ZDD	3.63	2.85	4.45	4.42
2GB1	4.88	5.19	6.48	8.72
2JZQ	8.46	6.62	9.62	9.93
2JUC	6.93	5.84	7.54	9.02
2MR9	5.11	5.17	7.42	7.97
2P5K	8.37	7.76	8.52	9.46
2P6J	6.54	5.43	10.82	6.8
2P81	4.64	3.77	6.43	8.37
2PMR	4.4	4.14	5.28	5.3
3V1A	3.01	2.32	3.96	4.15

[1] The structure with lowest RMSD in the archive *A*.

[2] Using hierarchical clustering in genotype space.

[3] Using hierarchical clustering in phenotype space.

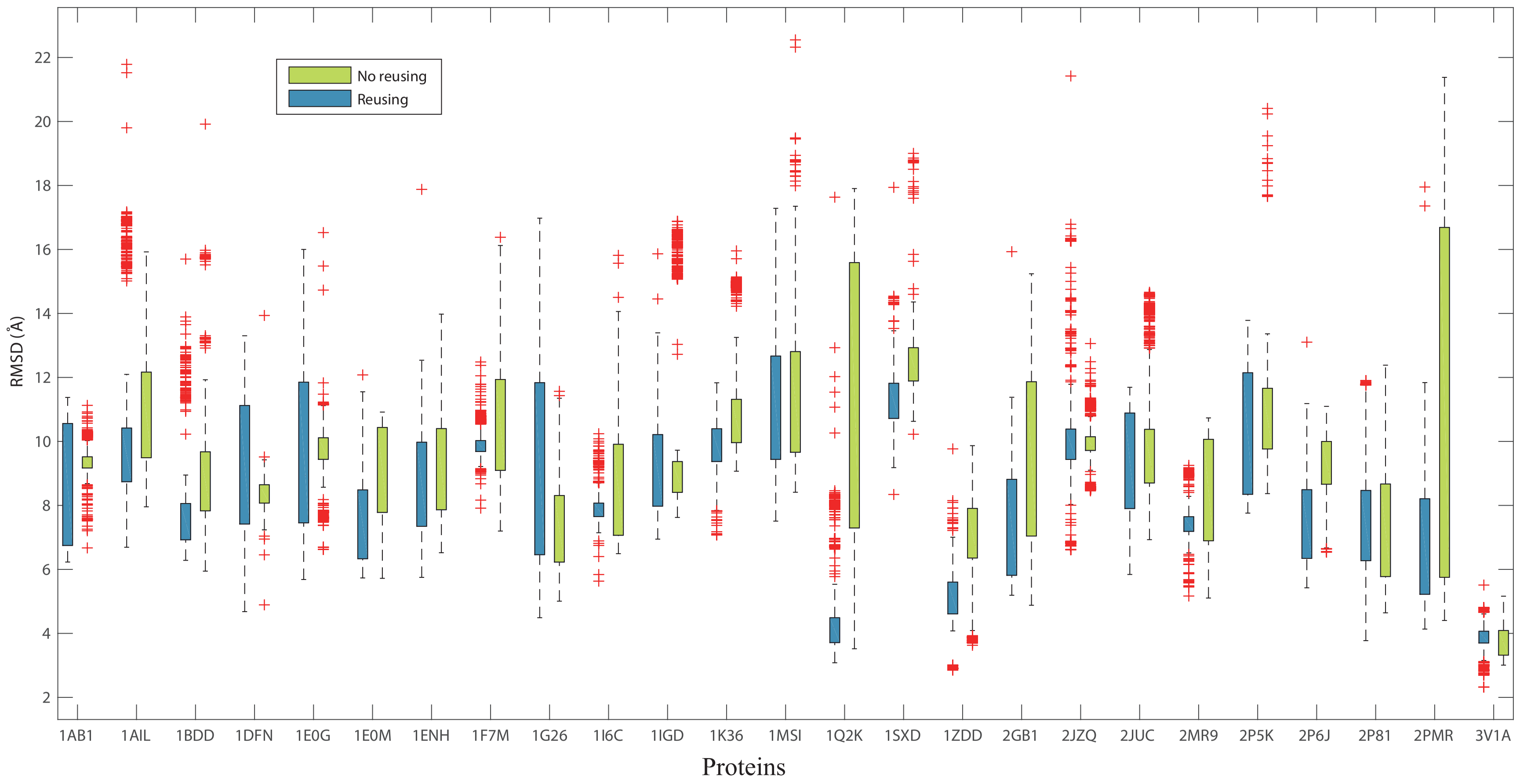


Figure 5.20: The box plot of RMSD$_{C_\alpha}$ of all decoy structures obtained by two strategy (reuse or not reuse search experience) for all target proteins.

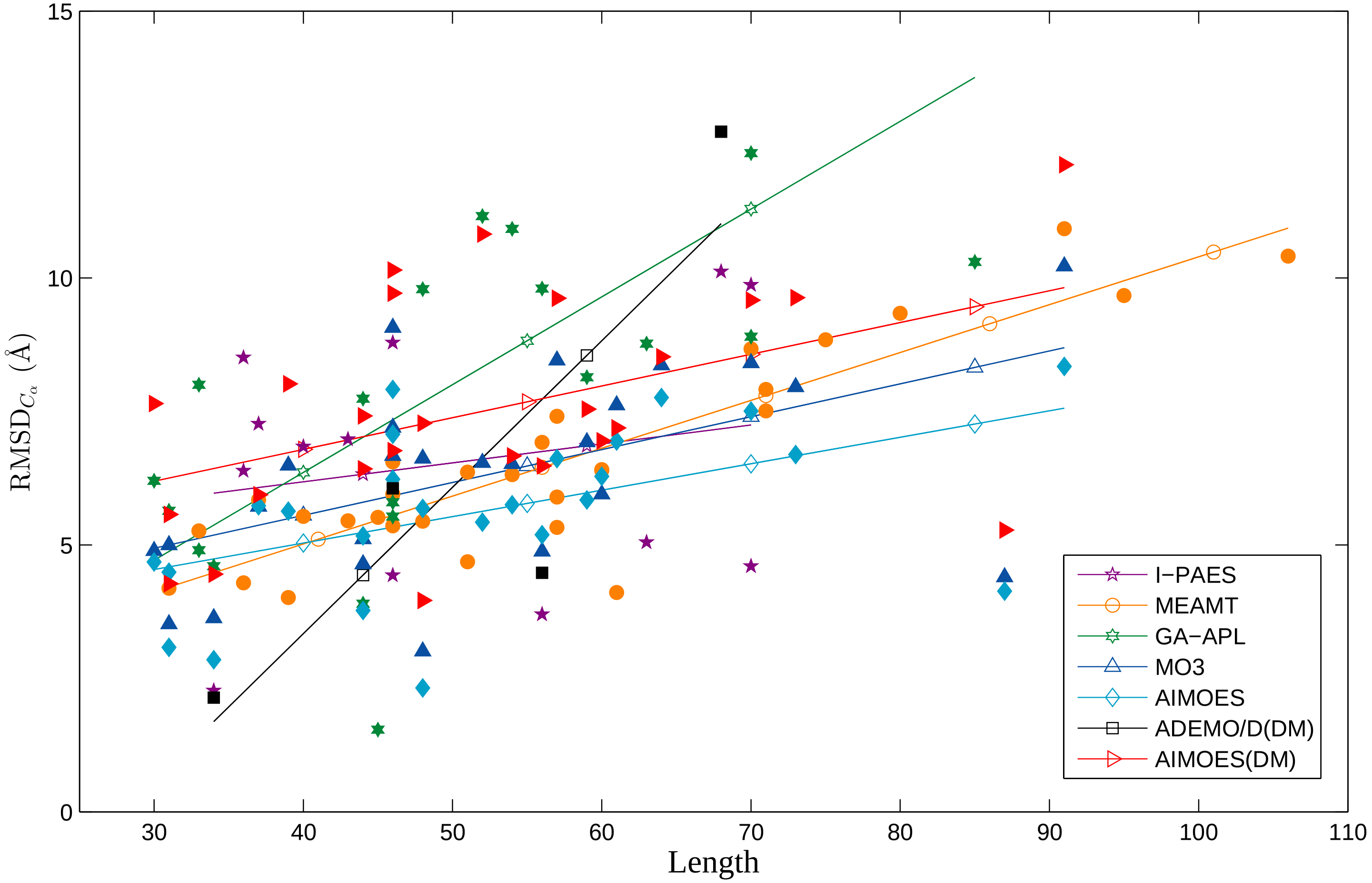


Figure 5.21: A qualitative comparison among different methods. A solid point represents a predicted result, and linear regression lines between protein length and RMSD$_{C_\alpha}$ for each method are plotted according to these solid points.

Table 5.3: Comparison of different methods using RMSD_{100}.

Method	num.pro	length range	class	evolution	avg.RMSD_{100}
I-PAES [25]	13	[34 70]	α,β,α/β	2.5*10E5	10.62
MEAMT [26]	30	[31 106]	α,β,α/β	5*10E5	9.41
GA-APL [71]	20	[30 85]	α,β,α/β	(24 hours)	12.62
MO3 [48]	25	[30 91]	α,β,α/β	8*10E4	9.79
AIMOES	25	[30 91]	α,β,α/β	8*10E4	8.78
ADEMO/D (DM) [138]	4	[34 68]	α,α/β	2.0*10E5	9.16
AIMOES (DM)	25	[30 91]	α,β,α/β	8*10E4	11.76

Chapter 6

Conclusion and future work

Since its introduction, the PSP problem has generally been treated as a single-objective optimization problem. Recently, modeling the PSP problem as a multi-objective optimization problem has become popular since the set of Pareto optimal solutions, taken as an ensemble, can provide a better answer to PSP than the optimal solution from single-objective optimization which is usually a single structure.

First, a multi-objective optimization approach (MO3) with three objective functions is proposed to achieve the best Pareto optimal sets never seen before. Considering the factor of solvent, we have for the first time incorporated SASA as the third objective function to compensate bond and non-bond energies. We utilize a multi-objective evolutionary algorithm to handle the problem. The performance of the method is verified by folding sixty-six proteins with sequence length of 14 345. The experimental results suggest that the method with three objectives is superior to the one with two objectives in terms of the required number of iterations and accuracy, and can generate better or competitive solutions compared with other prior methods. This result demonstrates that taking the effect of solvent into consideration is necessary and effective for handling the PSP problem. Such demonstration was not seen before to our best knowledge.

Second, driven by the motivation of theory and application, great efforts have been made to solving the PSP problem. However, it remains fascinating and challenging. Progress in FMs was slow, limited by the inaccuracies of energy function and the huge conformation search space. In the work of AIMOES, an enhanced multi-objective evolutionary algorithm is proposed to make effort for these two aspects. Recent work have showed that, modeling PSP as MOOP could provide more fruitful solutions and better answer than as single-objective optimization problem. This work followed the idea and treated PSP as a three-objective optimization problem. We decomposed the CHARMM22 into bond and non-bond energy as the first and second objective. Considering the solvent effect, SASA was incorporated as the third objective. Moreover, a elitism based multi-objective evolutionary algorithm was designed to execute conformation space searching. In order to improve the quality of search, a evolutionary scheme was incorporated to reusing the search experience by fetching the information stored in non-dominated solution archive. At last, a decision maker based on hierarchical clustering was proposed in the genotype and phenotype space. Twenty-five benchmark proteins were tested to verify the performance of the proposed method. The experiment results showed the power of the proposed method to solve PSP. We also compared it with four evolution computation algorithm for solving PSP, and a relatively fair comparison was carried out. The result suggested that the proposed method can obtain better or competitive results with them. It should be noted that, considering the time-consuming evolution times, our method seems more efficient.

In the future, we intend to apply these methods to more proteins to verify its performance. Theoretical analysis of selection operators and mutation operators in the algorithms should be performed because these operators contribute to the overall improvement of the method. In addition, we plan to include more objectives to improve the accuracy of the protein structure prediction because the energy landscape produced by the existing protein potential energy functions does not fit well with the real energy landscape. We should thus include other objective functions to amend the energy function. we will continue to pay attention to the energy function and confrontation search strategy. In these works, the objective functions are all physical. It is reasonable to incorporate the statistical energy function items, because they are more powerful for PSP confessedly. Moreover, other recent optimization method with advanced performance should be explored to enhance the conformation space search.

Bibliography

[1] Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.

[2] Lawrence A Kelley and Michael JE Sternberg. Protein structure prediction on the web: a case study using the phyre server. *Nature protocols*, 4(3):363, 2009.

[3] Ambrish Roy, Alper Kucukural, and Yang Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725, 2010.

[4] Yang Zhang. Progress and challenges in protein structure prediction. *Current opinion in structural biology*, 18(3):342–348, 2008.

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

[6] UniProt Consortium et al. Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169, 2017.

[7] Roshni Bhattacharya, Peter W Rose, Stephen K Burley, and Andreas Prlić. Impact of genetic variation on three dimensional structure and function of proteins. *PloS One*, 12(3):e0171355, 2017.

[8] Jeffrey Skolnick and Jacquelyn S Fetrow. From genes to protein structure and function: novel applications of computational approaches in the genomic era. *Trends in biotechnology*, 18(1):34–39, 2000.

[9] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins: Structure, Function, and Bioinformatics*, 84(S1):4–14, 2016.

[10] Marco Biasini, Stefan Bienert, Andrew Waterhouse, Konstantin Arnold, Gabriel Studer, Tobias Schmidt, Florian Kiefer, Tiziano Gallo Cassarino, Martino Bertoni, Lorenza Bordoli, et al. Swiss-model: modelling protein tertiary and quaternary structure using evolutionary information. *Nucleic Acids Research*, page gku340, 2014.

[11] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, and Yang Zhang. The i-tasser suite: protein structure and function prediction. *Nature Methods*, 12(1):7–8, 2015.

[12] Arunachalam Jothi. Principles, challenges and advances in ab initio protein structure prediction. *Protein and peptide letters*, 19(11):1194–1204, 2012.

[13] Kristian W Kaufmann, Gordon H Lemmon, Samuel L DeLuca, Jonathan H Sheehan, and Jens Meiler. Practically useful: what the rosetta protein modeling suite can do for you. *Biochemistry*, 49(14):2987–2998, 2010.

[14] Dong Xu and Yang Zhang. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function, and Bioinformatics*, 80(7):1715–1735, 2012.

[15] Richard Bonneau and David Baker. Ab initio protein structure prediction: progress and prospects. *Annual review of biophysics and biomolecular structure*, 30(1):173–189, 2001.

[16] Corey Hardin, Taras V Pogorelov, and Zaida Luthey-Schulten. Ab initio protein structure prediction. *Current opinion in structural biology*, 12(2):176–181, 2002.

[17] Michael Feig. Computational protein structure refinement: almost there, yet still so far to go. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 7(3), 2017.

[18] Md Kamrul Islam and Madhu Chetty. Clustered memetic algorithm with local heuristics for ab initio protein structure prediction. *IEEE Transactions on Evolutionary Computation*, 17(4):558–576, 2013.

[19] David Baker and Andrej Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, 2001.

[20] Lisa N Kinch, Wenlin Li, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Nick V Grishin. Evaluation of free modeling targets in casp11 and roll. *Proteins: Structure, Function, and Bioinformatics*, 84(S1):51–66, 2016.

[21] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[22] Jiahai Wang, Weiwei Zhang, and Jun Zhang. Cooperative differential evolution with multiple populations for multiobjective optimization. *IEEE Transactions on Cybernetics*, 46(12):2848–2861, 2016.

[23] Jiahai Wang, Ying Zhou, Yong Wang, Jun Zhang, CL Philip Chen, and Zibin Zheng. Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms. *IEEE Transactions on Cybernetics*, 46(3):582–594, 2016.

[24] Jiahai Wang, Jianjun Liao, Ying Zhou, and Yiqiao Cai. Differential evolution enhanced with multiobjective sorting-based mutation operators. *IEEE Transactions on Cybernetics*, 44(12):2792–2805, 2014.

[25] Vincenzo Cutello, Giuseppe Narzisi, and Giuseppe Nicosia. Computational studies of peptide and protein structure prediction problems via multiobjective evolutionary algorithms. In *Multiobjective Problem Solving from Nature*, pages 93–114. Springer, 2008.

[26] Christiane Regina Soares Brasil, Alexandre Claudio Botazzo Delbem, and Fernando Luís Barroso da Silva. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. *Journal of Computational Chemistry*, 34(20):1719–1734, 2013.

[27] Mohammad Reza Bonyadi and Zbigniew Michalewicz. Particle swarm optimization for single objective continuous space problems: a review, 2017.

[28] Shuangbao Song, Junkai Ji, Xingqian Chen, Shangce Gao, Zheng Tang, and Yuki Todo. Adoption of an improved pso to explore a compound multi-objective energy function in protein structure prediction. *Applied Soft Computing*, 72:539–551, 2018.

[29] Shangce Gao, Yirui Wang, Jiahai Wang, and JiuJun Cheng. Understanding differential evolution: A poisson law derived from population interaction network. *Journal of computational science*, 21:140–149, 2017.

[30] Junkai Ji, Shangce Gao, Shuaiqun Wang, Yajiao Tang, Hang Yu, and Yuki Todo. Self-adaptive gravitational search algorithm with a modified chaotic local search. *IEEE Access*, 5:17881–17895, 2017.

[31] Zhenyu Song, Shangce Gao, Yang Yu, Jian Sun, and Yuki Todo. Multiple chaos embedded gravitational search algorithm. *IEICE Transactions on Information and Systems*, 100(4):888–900, 2017.

[32] Yirui Wang, Yang Yu, Shangce Gao, Haiyu Pan, and Gang Yang. A hierarchical gravitational search algorithm with an effective gravitational constant. *Swarm and Evolutionary Computation*, 46:118–139, 2019.

[33] Junkai Ji, Shuangbao Song, Cheng Tang, Shangce Gao, Zheng Tang, and Yuki Todo. An artificial bee colony algorithm search guided by scale-free networks. *Information Sciences*, 473:142–165, 2019.

[34] Shi Wang, Shuangyu Song, Yang Yu, Zhe Xu, Hanaki Yachi, and Shangce Gao. Multiple chaotic cuckoo search algorithm. In *International Conference on Swarm Intelligence*, pages 531–542. Springer, 2017.

[35] Yang Yu, Shangce Gao, Shi Cheng, Yirui Wang, Shuangyu Song, and Fenggang Yuan. Cbso: a memetic brain storm optimization with chaotic local search. *Memetic Computing*, 10(4):353–367, 2018.

[36] Shangce Gao, Shuangbao Song, Jiujun Cheng, Yuki Todo, and Mengchu Zhou. Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(4):1365–1378, 2018.

[37] Shuangbao Song, Shangce Gao, Xingqian Chen, Dongbao Jia, Xiaoxiao Qian, and Yuki Todo. Aimoes: Archive information assisted multi-objective evolutionary strategy for ab initio protein structure prediction. *Knowledge-Based Systems*, 146:58–72, 2018.

[38] Zhenyu Song, Yajiao Tang, Xingqian Chen, Shuangbao Song, Shuangyu Song, and Shangce Gao. A preference-based multi-objective evolutionary strategy for ab initio prediction of proteins. In *2017 International Conference on Progress in Informatics and Computing (PIC)*, pages 7–12. IEEE, 2017.

[39] Junkai Ji, Shuangbao Song, Yajiao Tang, Shangce Gao, Zheng Tang, and Yuki Todo. Approximate logic neuron model trained by states of matter search algorithm. *Knowledge-Based Systems*, 163:120–130, 2019.

[40] Shangce Gao, MengChu Zhou, Yirui Wang, Jiujun Cheng, Hanaki Yachi, and Jiahai Wang. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE transactions on neural networks and learning systems*, (99):1–14, 2018.

[41] Shuangyu Song, Xingqian Chen, Cheng Tang, Shuangbao Song, Zheng Tang, and Yuki Todo. Training an approximate logic dendritic neuron model using

social learning particle swarm optimization algorithm. *IEEE Access*, 7:141947–141959, 2019.

[42] Jun Chin Ang, Andri Mirzal, Habibollah Haron, and Haza Nuzly Abdull Hamed. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. *IEEE/ACM transactions on computational biology and bioinformatics*, 13(5):971–989, 2016.

[43] Camila Silva de Magalhães, Diogo Marinho Almeida, Helio José Correa Barbosa, and Laurent Emmanuel Dardenne. A dynamic niching genetic algorithm strategy for docking highly flexible ligands. *Information Sciences*, 289:206–224, 2014.

[44] Esteban López-Camacho, María Jesús García Godoy, José García-Nieto, Antonio J Nebro, and José F Aldana-Montes. Solving molecular flexible docking problems with metaheuristics: A comparative study. *Applied Soft Computing*, 28:379–393, 2015.

[45] Yang Zhang, Daisuke Kihara, and Jeffrey Skolnick. Local energy landscape flattening: parallel hyperbolic monte carlo sampling of protein folding. *Proteins: Structure, Function, and Bioinformatics*, 48(2):192–201, 2002.

[46] Juyong Lee, Jinhyuk Lee, Takeshi N Sasaki, Masaki Sasai, Chaok Seok, and Jooyoung Lee. De novo protein structure prediction by dynamic fragment assembly and conformational space annealing. *Proteins: Structure, Function, and Bioinformatics*, 79(8):2403–2417, 2011.

[47] Philip Bradley, Kira MS Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, 2005.

[48] Shangce Gao, Shuangbao Song, Jiujun Cheng, Yuki Todo, and MengChu Zhou. Incorporation of solvent effect into multi-objective evolutionary algorithm for

improved protein structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP(99):1–1, 2017.

[49] Alexander D MacKerell Jr, Donald Bashford, MLDR Bellott, Roland Leslie Dunbrack Jr, Jeffrey D Evanseck, Martin J Field, Stefan Fischer, Jiali Gao, H Guo, Sookhee Ha, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The journal of Physical Chemistry B*, 102(18):3586–3616, 1998.

[50] Mark W Hauschild, Martin Pelikan, Kumara Sastry, and David E Goldberg. Using previous models to bias structural learning in the hierarchical boa. *Evolutionary Computation*, 20(1):135–160, 2012.

[51] Muhammad Iqbal, Will N Browne, and Mengjie Zhang. Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems. *IEEE Transactions on Evolutionary Computation*, 18(4):465–480, 2014.

[52] Sushil J Louis and John McDonnell. Learning with case-injected genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):316–328, 2004.

[53] L. Feng, Y. S. Ong, S. Jiang, and A. Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2017.

[54] C-L Hwang and Abu Syed Md Masud. *Multiple objective decision makingmethods and applications: a state-of-the-art survey*, volume 164. Springer Science & Business Media, 2012.

[55] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature PPSN VI*, pages 849–858. Springer, 2000.

[56] Carlos Segura, Carlos A Coello Coello, Gara Miranda, and Coromoto León. Using multi-objective evolutionary algorithms for single-objective optimization. *4OR*, 11(3):201–228, 2013.

[57] Bo Huang, Meng Chu Zhou, GongXuan Zhang, Ahmed Chiheb Ammari, Ahmed Alabdulwahab, and Ayman G Fayoumi. Lexicographic multiobjective integer programming for optimal and structurally minimal petri net supervisors of automated manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(11):1459–1470, 2015.

[58] Xingquan Zuo, Cheng Chen, Wei Tan, and Meng Chu Zhou. Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):1030–1041, 2015.

[59] Darrell F Lochtefeld and Frank W Ciarallo. An analysis of decomposition approaches in multi-objectivization via segmentation. *Applied Soft Computing*, 18:209–222, 2014.

[60] Carlos Segura, Eduardo Segredo, and Coromoto León. Scalability and robustness of parallel hyperheuristics applied to a multiobjectivised frequency assignment problem. *Soft Computing*, 17(6):1077–1093, 2013.

[61] Jiahai Wang, Ying Zhou, Yong Wang, Jun Zhang, CL Chen, and Zibin Zheng. Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms. *IEEE Transactions on Cybernetics*, 46(3):582–594, 2015.

[62] Hai-Peng Ren, Xiao-Na Huang, and Jia-Xuan Hao. Finding robust adaptation gene regulatory networks using multi-objective genetic algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(3):571–577, 2016.

[63] Y. Zhang, D. Gong, and J. Cheng. Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP(99):1–1, 2015.

[64] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using rosetta. *Methods in Enzymology*, 383:66–93, 2004.

[65] Sitao Wu, Jeffrey Skolnick, and Yang Zhang. Ab initio modeling of small proteins by iterative tasser simulations. *BMC Biology*, 5(1):17, 2007.

[66] Glennie Helles. A comparative study of the reported performance of ab initio protein structure prediction algorithms. *Journal of the Royal Society Interface*, 5(21):387–396, 2008.

[67] Daniel WA Buchan, Federico Minneci, Tim CO Nugent, Kevin Bryson, and David T Jones. Scalable web services for the psipred protein analysis workbench. *Nucleic Acids Research*, 41(W1):W349–W357, 2013.

[68] Pierrick Craveur, Agnel Praveen Joseph, Pierre Poulain, Alexandre G de Brevern, and Joseph Rebehmed. Cis–trans isomerization of omega dihedrals in proteins. *Amino Acids*, 45(2):279–289, 2013.

[69] Roland L Dunbrack. Rotamer libraries in the 21 st century. *Current Opinion in Structural Biology*, 12(4):431–440, 2002.

[70] José C Calvo, Julio Ortega, and Mancia Anguita. Pitagoras-psp: Including domain knowledge in a multi-objective approach for protein structure prediction. *Neurocomputing*, 74(16):2675–2682, 2011.

[71] Bruno Borguesan, Mariel Barbachan e Silva, Bruno Grisci, Mario Inostroza-Ponta, and Márcio Dorn. Apl: An angle probability list to improve knowledge-based metaheuristics for the three-dimensional protein structure prediction. *Computational Biology and Chemistry*, 59:142–157, 2015.

[72] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[73] Adam Zemla, Česlovas Venclovas, John Moult, and Krzysztof Fidelis. Processing and evaluation of predictions in casp4. *Proteins: Structure, Function, and Bioinformatics*, 45(S5):13–21, 2001.

[74] Adam Zemla. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

[75] Cyrus Levinthal. Are there pathways for protein folding. *J. Chim. phys*, 65(1):44–45, 1968.

[76] Philip Bradley, Kira M. S. Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, 2005.

[77] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[78] William E Hart and Sorin Istrail. Robust proofs of np-hardness for protein folding: general lattices and energy potentials. *Journal of Computational Biology*, 4(1):1–22, 1997.

[79] Jooyoung Lee, Sitao Wu, and Yang Zhang. *Ab initio protein structure prediction*, chapter 1, pages 3–25. Springer, 2009.

[80] Bernard R Brooks, Charles L Brooks, Alexander D MacKerell, Lennart Nilsson, Robert J Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christian Bartels, Stefan Boresch, et al. Charmm: the biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614, 2009.

[81] William L Jorgensen and Julian Tirado-Rives. The opls [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society*, 110(6):1657–1666, 1988.

[82] Yuedong Yang and Yaoqi Zhou. Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins: Structure, Function, and Bioinformatics*, 72(2):793–803, 2008.

[83] Jian Zhang and Yang Zhang. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PLoS One*, 5(10):e15386, 2010.

[84] Joshua D Knowles, Richard A Watson, and David W Corne. Reducing local optima in single-objective problems by multi-objectivization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 269–283. Springer, 2001.

[85] Soo-Yong Shin, In-Hee Lee, Dongmin Kim, and Byoung-Tak Zhang. Multiobjective evolutionary optimization of dna sequences for reliable dna computing. *IEEE Transactions on Evolutionary Computation*, 9(2):143–158, 2005.

[86] Silvia Curteanu, Florin Leon, and Dan Gâlea. Alternatives for multiobjective optimization of a polymerization process. *Journal of Applied Polymer Science*, 100(5):3680–3695, 2006.

[87] Mehmet Kaya, Abdullah Sarhan, and Reda Alhajj. Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. *Computer Methods and Programs in Biomedicine*, 114(1):38–49, 2014.

[88] CA Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, 2006.

[89] Themis Lazaridis and Martin Karplus. Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology*, 10(2):139–145, 2000.

[90] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules j. am. chem. soc. 1995, 117, 5179-5197. *Journal of the American Chemical Society*, 118(9):2309–2309, 1996.

[91] Bernard Brooks and Martin Karplus. Harmonic dynamics of proteins: normal modes and fluctuations in bovine pancreatic trypsin inhibitor. *Proceedings of the National Academy of Sciences*, 80(21):6571–6575, 1983.

[92] Vincenzo Cutello, Giuseppe Narzisi, and Giuseppe Nicosia. A multi-objective evolutionary approach to the protein structure prediction problem. *Journal of The Royal Society Interface*, 3(6):139–151, 2006.

[93] Sandra M Scós Venske, Richard A Gonçalves, Elaine M Benelli, and Myriam R Delgado. A multiobjective algorithm for protein structure prediction using adaptive differential evolution. In *Intelligent Systems (BRACIS), 2013 Brazilian Conference on*, pages 263–268. IEEE, 2013.

[94] David Becerra, Angelica Sandoval, Daniel Restrepo-Montoya, and Luis Fernando Nino. A parallel multi-objective ab initio approach for protein structure prediction. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 137–141. IEEE, 2010.

[95] MV Judy, KS Ravichandran, and K Murugesan. A multi-objective evolutionary algorithm for protein structure prediction with immune operators. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(4):407–413, 2009.

[96] Byungkook Lee and Frederic M Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379IN3–400IN4, 1971.

[97] Eshel Faraggi, Tuo Zhang, Yuedong Yang, Lukasz Kurgan, and Yaoqi Zhou. Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of Computational Chemistry*, 33(3):259–267, 2012.

[98] Martin Karplus and Eugene Shakhnovich. Protein folding: theoretical studies of thermodynamics and dynamics. *Protein Folding*, pages 127–195, 1992.

[99] Themis Lazaridis, Georgios Archontis, and Martin Karplus. Enthalpic contribution to protein stability: insights from atom-based calculations and statistical mechanics. *Advances in Protein Chemistry*, 47:231–306, 1995.

[100] William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983.

[101] Themis Lazaridis and Martin Karplus. Effective energy function for proteins in solution. *Proteins: Structure, Function, and Bioinformatics*, 35(2):133–152, 1999.

[102] Sergio A Hassan, Ernest L Mehler, Daqun Zhang, and Harel Weinstein. Molecular dynamics simulations of peptides and proteins with a continuum electrostatic model based on screened coulomb potentials. *Proteins: Structure, Function, and Bioinformatics*, 51(1):109–125, 2003.

[103] Laura Wesson and David Eisenberg. Atomic solvation parameters applied to molecular dynamics of proteins in solution. *Protein Science*, 1(2):227–235, 1992.

[104] David Eisenberg and Andrew D McLachlan. Solvation energy in protein folding and binding. *Nature*, 319(6050):199–203, 1985.

[105] Di Qiu, Peter S Shenkin, Frank P Hollinger, and W Clark Still. The gb/sa continuum model for solvation. a fast analytical method for the calculation of approximate born radii. *The Journal of Physical Chemistry A*, 101(16):3005–3014, 1997.

[106] Andreas Klamt and GJGJ Schüürmann. Cosmo: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *Journal of the Chemical Society, Perkin Transactions 2*, 5:799–805, 1993.

[107] Christoph Hartlmüller, Christoph Göbl, and Tobias Madl. Prediction of protein structure using surface accessibility data. *Angewandte Chemie*, 128(39):12149–12153, 2016.

[108] Ingo Rechenberg. Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment*, page Library translation No. 1122, 1965.

[109] H-P Schwefel. *Evolutionsstrategie und numerische Optimierung*. Technische Universität Berlin, 1975.

[110] Joshua Knowles and David Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages 98–105. IEEE, 1999.

[111] Joshua D Knowles and David W Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[112] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions on*, 3(2):82–102, 1999.

[113] Giovanni Stracquadanio and Giuseppe Nicosia. Computational energy-based redesign of robust proteins. *Computers & Chemical Engineering*, 35(3):464–473, 2011.

[114] Michiel JL De Hoon, Seiya Imoto, John Nolan, and Satoru Miyano. Open source clustering software. *Bioinformatics*, 20(9):1453–1454, 2004.

[115] Fionn Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The computer journal*, 26(4):354–359, 1983.

[116] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.

[117] Robin C Purshouse and Peter J Fleming. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In *Evolutionary multi-criterion optimization*, pages 16–30. Springer, 2003.

[118] Sandra M Venske, Richard A Gonçalves, Elaine M Benelli, and Myriam R Delgado. Ademo/d: An adaptive differential evolution for protein structure prediction problem. *Expert Systems with Applications*, 56:209–226, 2016.

[119] Márcio Dorn, Luciana S Buriol, and Luis C Lamb. A hybrid genetic algorithm for the 3-d protein structure prediction problem using a path-relinking strategy. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2709–2716. IEEE, 2011.

[120] Márcio Dorn and Osmar Norberto de Souza. Cref: a central-residue-fragment-based method for predicting approximate 3-d polypeptides structures. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1261–1267. ACM, 2008.

[121] Giuseppe Nicosia and Giovanni Stracquadanio. Generalized pattern search and mesh adaptive direct search algorithms for protein structure prediction. In *International Workshop on Algorithms in Bioinformatics*, pages 183–193. Springer, 2007.

[122] Angelo Marcello Anile, Vincenzo Cutello, Giuseppe Narzisi, Giuseppe Nicosia, and Salvatore Spinella. Determination of protein structure and dynamics combining immune algorithms and pattern search methods. *Natural Computing*, 6(1):55–72, 2007.

[123] B Jayaram, Kumkum Bhushan, Sandhya R Shenoy, Pooja Narang, Surojit Bose, Praveen Agrawal, Debashish Sahu, and Vidhu Pandey. Bhageerath: an energy based web enabled computer software suite for limiting the search space of tertiary structures of small globular proteins. *Nucleic Acids Research*, 34(21):6195–6204, 2006.

[124] Lee R Cooper, David W Corne, and M James C Crabbe. Use of a novel hill-climbing genetic algorithm in protein folding simulations. *Computational Biology and Chemistry*, 27(6):575–580, 2003.

[125] J David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100. L. Erlbaum Associates Inc., 1985.

[126] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[127] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies. *Natural Computing*, 1(1):3–52, 2002.

[128] JL Klepeis, MJ Pieja, and CA Floudas. Hybrid global optimization algorithms for protein structure prediction: Alternating hybrids. *Biophysical Journal*, 84(2):869–882, 2003.

[129] Yang Zhang, Andrzej Kolinski, and Jeffrey Skolnick. Touchstone ii: a new approach to ab initio protein structure prediction. *Biophysical journal*, 85(2):1145–1164, 2003.

[130] Julian Lee, Seung-Yeon Kim, and Jooyoung Lee. Protein structure prediction based on fragment assembly and parameter optimization. *Biophysical Chemistry*, 115(2):209–214, 2005.

[131] George Chikenji, Yoshimi Fujitsuka, and Shoji Takada. A reversible fragment assembly method for de novo protein structure prediction. *The Journal of Chemical Physics*, 119(13):6895–6903, 2003.

[132] Mandavilli Srinivas and Lalit M Patnaik. Genetic algorithms: A survey. *Computer*, 27(6):17–26, 1994.

[133] Samir W Mahfoud. Crowding and preselection revisited. *Urbana*, 51:61801, 1992.

[134] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2017.

[135] Eckart Ziztler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design, Optimization, and Control*, pages 95–100, 2002.

[136] Andriy Kryshtafovych, Alessandro Barbato, Krzysztof Fidelis, Bohdan Monastyrskyy, Torsten Schwede, and Anna Tramontano. Assessment of the assessment: evaluation of the model quality estimates in casp10. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):112–126, 2014.

[137] Christine A Orengo, AD Michie, S Jones, David T Jones, MB Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[138] Sandra M. Venske, Richard A. Gonalves, Elaine M. Benelli, and Myriam R. Delgado. Ademo/d: An adaptive differential evolution for protein structure prediction problem. *Expert Systems with Applications*, 56:209 – 226, 2016.

[139] Steven A Combs, Samuel L DeLuca, Stephanie H DeLuca, Gordon H Lemmon, David P Nannemann, Elizabeth D Nguyen, Jordan R Willis, Jonathan H Sheehan, and Jens Meiler. Small-molecule ligand docking into comparative models with rosetta. *Nature Protocols*, 8(7):1277–1298, 2013.

[140] Oliviero Carugo and Sándor Pongor. A normalized root-mean-spuare distance for comparing protein three-dimensional structures. *Protein Science*, 10(7):1470–1473, 2001.