Natural Computing
https://doi.org/10.1007/s11047-022-09916-z

A self-adaptive evolutionary algorithm using Monte Carlo Fragment insertion and conformation clustering for the protein structure prediction problem

Rafael Stubs Parpinelli[1] · Nilcimar Neitzel Will[1] · Renan Samuel da Silva[1]

Accepted: 27 July 2022

Abstract
The Protein Structure Prediction Problem is one of the most important and challenging open problems in Computer Science and Structural Bioinformatics. Accurately predicting protein conformations would significantly impact several fields, such as understanding proteinopathies and developing smart protein-based drugs. As such, this work has as its primary goal to improve the prediction power of *ab initio* methods by utilizing a self-adaptive evolutionary algorithm using Monte Carlo based fragment insertion and conformational clustering. A meta-heuristic is used as the core of the conformation sampling process with fragment insertion, feeding domain-specific information into the process. The online parameter control routines allow the method to adapt to a protein's structure specificity and behave dynamically in different stages of the optimization process. The results obtained by the proposed method were compared to results obtained from several other algorithms found in the literature. It is possible to conclude that the proposed method is highly competitive in terms of free-energy and RMSD for the protein set used in the experiments.

Keywords Bioinformatics · Fragment insertion · Protein structure prediction problem · ab initio method · Clustering

1 Introduction

Proteins are responsible for several vital functions, such as structural, metabolic, and regulatory. Moreover, the protein function is directly dependent on its three-dimensional conformation (Kihara 2014). Therefore, knowing the protein conformation can give insight into specific proteins' function and help understand and treat diseases caused by misfolded proteins, such as Parkinson's and other diseases (Walsh 2002).

The Protein Structure Prediction Problem (PSPP) is considered one of the leading open problems in Computer Science and Structural Bioinformatics (Dorn et al. 2014; Lopes 2008). The PSPP consists of computationally finding the tertiary structure of a protein-based on its respective primary sequence. A plethora of models has been proposed over the years with different levels of biological plausibility, complexity, and accuracy. One of the simplest models, the on-lattice 2D HP is an $\mathcal{NP}$-hard problem (Berger and Leighton 1998). The $\mathcal{NP}$-hardness of the PSPP has been generalized for all on-lattice models (Hart and Istrail 1997) and all off-lattice models. Nevertheless, it is reasonable to expect more complex models to be $\mathcal{NP}$-hard too.

Recently, the AlphaFold, the deep learning algorithm developed by DeepMind, achieved highly promising results in the CASP13 and CASP14 competition (Jumper et al. 2021). The AlphaFold combines features derived from homologous templates and from multiple sequence alignment to generate the predicted structure. Nevertheless, AlphaFold has some drawbacks, such as the bias to the

✉ Rafael Stubs Parpinelli
rafael.parpinelli@udesc.br

Nilcimar Neitzel Will
nil-cc@yahoo.com.br

Renan Samuel da Silva
uber.renan@gmail.com

[1] Graduate Program in Applied Computing, State University of Santa Catarina, Joinville, Brazil

Published online: 11 August 2022

Protein Data Bank (PDB) database, and the heavy dependency on high-computational efforts for training their model (David et al. 2022). Also, given the number of possible natural and artificial protein structures, it is currently unfeasible to rely on template-based methods to predict any unknown structure with consistent quality. As such, the PSPP is still considered to be an open problem. So far, there is no viable general solution to this challenging problem. In this way, metaheuristics can be a faster option to the problem even though achieved results are not the same as AlphaFold.

As shown in Nunes et al. (2016), it is possible to solve the 2D HP problem to optimality for small instances of the problem. However, the presented model has small biological plausibility and suffers from poor scalability. To approach a full atomic representation of the problem, the use of meta-heuristics becomes the primary approach for attempting to solve the problem. Several works in the literature have tried to solve the PSPP using meta-heuristics. In Borguesan et al. (2015), a genetic algorithm is used. -Garza-Fabre et al. (2016) utilized a memetic algorithm. A simulated annealing approach is proposed in Silva and Parpinelli (2018). In Silva and Parpinelli (2019), the use of differential evolution is presented.

It is well known that the performance of meta-heuristics is highly dependent on the set of parameters utilized during the optimization (Karafotias et al. 2015), and for solving the PSPP, it is no different. Most of the classical approaches using meta-heuristics have a fixed set of parameters that control the optimization process's behavior deterministically. These parameters must be found *a priori* and usually require a slow and computationally intensive process such as a grid search. Furthermore, it is unreasonable to expect that a single set of parameters will have optimal performance over a broad set of instances of a problem, especially for a highly complex problem such as the PSPP. For these reasons, many authors have employed the use of on-line parameter control. That is, actively monitoring and changing the parameters throughout the optimization (Parpinelli et al. 2019). This approach not only allows the optimizer to adapt to different instances of a problem, but it also permits that the optimizer adapts to different regions of the energy landscape during the optimization process of a single instance.

Given the problem's high complexity, a blind optimizer might underperform due to the energy function's roughness and the ample search space. In this case, it is possible to employ a hybrid algorithm (Blum et al. 2011). A hybrid algorithm allows the optimization procedure to be guided to search for more relevant directions, thus improving overall performance. Given the high dimensionality of the problem and its multimodality, this becomes a requirement to improve the overall prediction quality. This is achieved by employing fragment insertion during the optimization process, which directly applies small structures with biological plausibility into the conformation being predicted.

Based on Anfisen's hypothesis, the point of the lowest potential energy of the conformational search space will correspond to the native conformation. Given the size of the search space, finding this point is a non-trivial task. However, since the energy landscape tends to have a funnel-like shape, one can leverage the distribution of multiple predictions in the hope of finding the overall direction of the native conformation. Hence, the clusterization of predicted conformations can be employed to this end. There is a tendency that the more prominent clusters will be closer to the native conformation.

This work has as its primary goal to study and attempt to solve the PSPP. For this, an off-lattice *ab initio* method will be used. The proteins will be represented computationally as torsion angles of the backbone and side-chain centroids. The *ab initio* model will be optimized using a hybrid optimizer based on a self-adaptive evolutionary algorithm with Monte Carlo (MC) and Local Search, named PPF-MC. Moreover, a conformational clustering routine is employed to identify promising regions of the search space. The final clusters are then subject to the Hooke-Jeeves local search procedure and then fed into a repacking procedure to translate the model from a centroid one into a full atom configuration.

The organization of this work is: Sect. 2 presents the PSPP; Sect. 3 discusses related works; Sect. 4 presents the proposed method; Sect. 5 presents the experimental setup and results obtained; finally, Sect. 6 presents the conclusions and future work directions.

2 The protein structure prediction problem

The primary structure of the protein is considered the linear sequence of amino acids within a protein. Proteins are built from a set of amino acids, each of which has a unique side-chain composed of different chemistries. The PSPP consists of taking the primary sequence of a protein as input and outputting a prediction of its native conformation (or tertiary structure). There are several methods for doing so, with varying degrees of success. These methods are categorized into *ab initio* and knowledge-based, depending on how the method operates. The knowledge-based methods, represented by Homology Modeling (HM) and Threading Modeling (TM), are the two methods with the best results so far. However, they rely on the existence of protein with a known conformation that has a high degree of homology or on the existence of a suitable template of good quality.

Another class of protein prediction algorithms is the *ab initio* methods (Lee et al. 2017). From the Latin,

ab initio means *first principles*. The information about the physical and chemical properties of the proteins are encoded on the protein representation. Its interactions are evaluated by energy functions (or scoring functions). A given protein *in natura* seeks its point of least potential energy (Anfinsen 1973). Based on this fact, an energy function for an *ab initio* method tries to evaluate the potential energy of a given conformation. With a search procedure, it is possible to find its point of least potential energy, which should be close to its respective native conformation. In other words, *ab initio* methods can be seen as an optimization problem where the objective function is the energy function of the protein, and its variables are the degrees of freedom from its computational representation. Also, *ab initio* methods can be classified into on-lattice and off-lattice models.

On-lattice models consist of a protein representation where a lattice bounds its shape. Despite the on-lattice model's simplicity, it still is an intractable problem on a large scale due to its $\mathcal{NP}$-completeness. Nevertheless, from a practical point of view, these models can not represent a protein with enough details. Therefore, a more robust representation is required. This is possible using off-lattice models that are models not constrained by a lattice.

The AB model can be considered the simplest off-lattice model, in which the amino acids are represented as spheres that are either hydrophobic or polar, and the angles between them are not constrained (Berger and Leighton 1998; Boiani and Parpinelli 2020). A more detailed model consists of using the coordinates of the C_α. In this model, the amino acids are abstracted into spheres; however, they maintain their properties such as polarity and hydrophobicity, allowing for an increased level of detail. It is possible to represent each of the amino acid heavy atoms (Nitrogen, Carbon, and Oxygen) individually, instead of using a sphere to represent the whole amino acid. This model allows for interactions between individual atoms in the backbone to be considered during the prediction. The protein backbone is what holds a protein together and gives its tertiary structure.

Increasing the level of detail, it is possible to represent all atoms in the backbone, including the hydrogen atoms. This permits that hydrogen bonds be taken into account, which plays a significant role in forming secondary structures and their interactions. The secondary structure refers to regular, recurring arrangements in the space of adjacent amino acid residues in a polypeptide chain. Furthermore, it is possible to include a centroid (also called ellipsoid) to describe the amino acid side-chain. With this, each amino acid's shape can be considered when predicting the protein's three-dimensional structure.

There are two main models to fully represent the protein: The backbone and side-chain torsion angles, and all atoms coordinates (Rohl et al. 2004). The former describes all atoms in the protein, including the side-chain. However, the bond length between these atoms is fixed and the position of the hydrogen atoms. This model is enough to describe the protein very accurately and take into account most of its interactions. Nevertheless, since artificial constraints are imposed in the model, it is possible that some proteins can not be correctly predicted because it depends on one of the aspects that the model abstracted. For this reason, the all-atom coordinates model can be employed. In this model, all atoms are represented and can have a degree of freedom. Currently, this model is the most accurate one, at the expense of adding up a hundred variables per amino acid.

Several energy functions in the literature allow for an *ab initio* approach, such as the AMBER, GROMOS, CHARMM, and Rosetta. The AMBER (Salomon-Ferrer et al. 2013) package contains a set of scoring functions based only on the potential energy of proteins (and other molecules). It was originally designed for molecular dynamics simulations. However, it is possible to use it to score a given conformation. Another package that offers energy functions for proteins is CHARMM (Brooks et al. 2009). Like AMBER, it is primarily intended for molecular dynamics simulations for several organic molecules of interest. Nevertheless, its scoring can be used for *ab initio* methods. The GROMOS package (Eichenberger et al. 2011) also focuses primarily on molecular dynamics simulations and provides energy scoring of protein conformations. A more detailed discussion of the energy functions is available at Dorn et al. (2014). In Narloch and Parpinelli (2016), the authors explored the differences between AMBER, CHARMM, and Rosetta. A further discussion on energy fields (from molecular dynamic packages) can be found in Vlachakis et al. (2014).

The Rosetta Suite (Rohl et al. 2004; Kaufmann et al. 2010) contains multiple energy functions for all-atom coordinates models, backbone and side-chain torsion angles, and backbone torsion angles with centroids for the side-chains. This suite also allows for the customization and creation of new energy functions. The Rosetta energy functions consider the protein's physicochemical properties and its statistical nature, based on a knowledge database with propensities of each amino acids. Information regarding the compactness of the structure and other properties, such as the formation of side-chain structures, are also computed. This removes the possibility of scoring the protein with a physical unit of measurement. Instead, the energy functions are measured by the Rosetta Energy Units (REU). More information about the Rosetta energy function is available in Alford et al. (2017).

With an energy function for scoring the protein conformations, it is possible to employ an optimizer to search for the least potential energy structures. This procedure must sample the conformation space accessible from the computational representation of a given protein. A vast range of methods has been employed over the years in the literature. In Li and Scheraga (1987), a Monte Carlo based search, is used to optimize a set of dihedral angles. Basin-hopping is a method where a random perturbation is applied to the conformation, and then a hill-climbing type of algorithm is employed to find a local minimum. It has been used in Prentiss et al. (2008) and Olson and Shehu (2012).

One particular branch of algorithms that have been used extensively in the literature is bio-inspired algorithms. The well-established algorithms are present in the literature, such as the Particle Swarm Optimization (Geng and Shen 2017), Differential Evolution (Hao et al. 2017), and the Genetic Algorithm (Higgs et al. 2010). Other algorithms that are not so widely used have also been explored for the PSPP, such as the Cuckoo Search (CS) (Ramyachitra and Ajeeth 2017) and the Bee Colony Algorithm (Li et al. 2015).

2.1 The Rosetta suite

Practically, working with an *ab initio* method is very time-consuming due to the high amount of boilerplate code required to model the protein and its molecular dynamics from scratch. Furthermore, this process is also very error-prone. The Rosetta Suite (Rohl et al. 2004) introduces a robust and validated suite for working with proteins and other macromolecules. It also includes multiple utilities for manipulating, pre-processing, and post-processing the protein conformations in a pipeline. The Rosetta Suite is free for academic use, and it is open-source [1].

One of the tools available at Rosetta and required for this work is fragment insertion. A fragment consists of a sequence of contiguous amino acids at a specific configuration extracted from some known structure protein. This sequence must fully match some continuous sequence of amino acids in the target protein, where the structure is unknown. The purpose of this is to use multiple fragments as building blocks. It is worth noting that homologous structures' fragments must be removed to avoid potential sampling from the same protein from another organism.

Creating a set of fragments for a particular target protein must be run only once per target and per-fragment size. Two sizes of fragments commonly used are 3 and 9. The fragment picker is responsible for searching a database of non-redundant protein conformations and sampling it to assemble fragments. More information about the inner workings of the Rosetta Fragment Picker is available in Gront et al. (2011).

Rosetta Suite includes two fragment insertion operators (called *movers*). One is found in Rosetta as *ClassicFragmentInsertion*, and the other operator is the smooth, found in Rosetta as *SmoothFragmentInsertion*.

The classical operator replaces one portion of the protein with its respective fragment. This change can be very aggressive and have a high changing impact on protein conformation.

The smooth fragment insertion applies a classic fragment insertion followed by a second fragment insertion that tries to minimize the Gunn Cost (Gunn 1997). The Gunn Cost measures the amount of change in a conformation due to the arm lever effect. The further away from the insertion point an amino acid is, the more it will move. Since the smooth fragment insertion tends to negate some of the change, it will preserve some of the protein structure, leading to a more localized change in the conformation. It is worth noting that the smooth fragment insertion is an optimization problem that minimizes the Gunn Cost. Since this operator tries to minimize the amount of change by its application, the impact on the energy score will be smaller, leading to smaller and more progressive changes.

3 Related works

This section provides a comparison with the most recent works in the literature. Only works using ab-initio with all-atom modeling or all backbone atoms with side-chains centroids are considered. The papers are presented in order of year of publication, starting from the year of 2015.

In Sudha et al. (2015), the authors applied SaDE with a diversity control strategy and local search operators to the PSPP. This modified version of SaDE, called DCSaDE-LS, was then compared with SaDE and other competing methods. DCSaDE-LS was able to outperform the other methods. However, it is worth noting that the authors used Met-enkephalin (1PLW), a small protein comprising only five amino acids, for the testing.

In Borguesan et al. (2015), the authors present a GA and a PSO use of APL. Both GA and PSO consider the statistical distribution of dihedral angles in the search operators. The use of that information leads to a significant boost in prediction performance.

An approach based on a Memetic algorithm is presented in Garza-Fabre et al. (2016). It uses fragment assembly as a local search form, and a particular type of crossover is employed a well. This new crossover operator operates exchanging loop regions between two-parent proteins to generate new offspring. In general, α-helix and β-sheets are

[1] Rosetta Suite available at https://www.rosettacommons.org/

relatively stable, predictable, and well-defined structures. Meanwhile, loop regions on the protein are highly unpredictable and have a high impact on the protein conformation. Therefore, one of this work's main contributions is operators specific for loop regions and a local search operator in a memetic environment.

Another memetic approach is presented in Correa et al. (2016). There, a multi-population based GA is employed in order to maintain diversity. Special crossover operators enforce and apply secondary structures to specific parts of the protein to more easily assemble them during the optimization process. A local operator is also employed in order to exploit possible suitable conformations. These operators replace random dihedral angles in the protein with angles gathered from APL (Borguesan et al. 2015), a database with statistical data about dihedral angle distributions. Also, a simulated annealing algorithm is utilized after the angle exchange to explore the conformation neighborhood.

In Narloch and Parpinelli (2017), the authors explored the use of different operators throughout the optimization process to maintain diversity and control exploration/exploitation. The results do point out that maintaining diversity and controlling exploitation/exploration improves the prediction results. The diversification point is explored in further detail in Narloch and Parpinelli (2016) and the point about controlling exploration/diversification is confirmed in Simoncini et al. (2017).

A method based on a variant of the EDA is proposed in Hao and Zhang (2017). In this work, the authors utilize a variant of EDA that works based on the energy distribution of the conformational energy and the acceptance rate of fragment insertions. The conformations with better energy are sampled more often, and the search is focused on the areas with a lower acceptance rate. This leads to a useful sampling of conformations while avoiding spending a high amount of function evaluations on regions with a high acceptance rate (such as α-helix) and focusing on less stable regions (such as coils and loops) that have a more significant effect on the conformation.

A DE approach is studied in Hao et al. (2017), where the authors employed the use of multiple sub-populations. Each sub-population is based on a cluster constructed using a feature reduction method. With this, similar solution vectors can be found, and then operators can be applied to similar individuals to intensify the search. Extra cluster operators are also applied and help maintain diversity and share information between clusters.

In de Oliveira et al. (2017), the authors argue that sequential sampling leads to better efficiency in terms of function evaluations and a better prediction. The sequential sampling based on fragment insertion follows the direction that proteins are assembled naturally. A reverse order sampling method and a non-sequential one are also explored. A key difference in this work is that conformational sampling starts with a small number of amino acids, adding new amino acids over the run.

In Oliveira et al. (2017), a method named SADE-SPL is proposed. The authors process the PDB base searching for structural patterns for coils and loops in knows structure proteins. The result of this search, called SPL, is then used as a domain-specific operator in SaDE. This work successfully utilized SaDE in a hybrid environment. However, due to the data-mining approach to finding structural patterns, this work might be considered a mix of an *ab initio* method with some characteristics of thread modeling, namely, the use of templates (structural patterns).

A proposal based on the Genetic Algorithm is presented in Borguesan et al. (2018). It uses a Restricted Tournament Selection to focus the crossover operations on individuals of at least a certain degree of similarity. This helps to maintain diversity and to cluster similar individuals. This work also uses a specialized fragment library, which focuses on inserting fragments covering adjacent sections of the same secondary structure. The GA operators also use NIAS information, an APL derivative, which feeds information into the sampling process.

A multi-stage strategy is utilized in Silva and Parpinelli (2018), where the authors presented the Multistage Simulated Annealing (MSA), which applies five different SA runs sequentially to a model with increasing levels of detail. Initially, a very rough model is constructed, and it is refined until a full-atom configuration is given as an output.

In Kandathil et al. (2018), the authors proposed two changes to the Rosetta *ab initio* protocol. The first consists of using a Bilevel Optimization (Sinha et al. 2018). As pointed by the authors, this approach had a sub-par result. Another proposal was the use of ILS. With ILS, forced perturbations are employed for loop regions. The use of ILS inside Rosetta's *ab initio* protocol leads to an improvement in prediction power. The authors also note the negative impact that mispredicted secondary structures can have on the tertiary structure prediction.

The use of the Artificial Bee Colony (ABC) metaheuristic is presented in Correa et al. (2018). Both the standard version of ABC with only minor improvements and a modified version (MOD-ABC) is availed. The modified version focuses the search on coils, loops, and turn regions. Both versions use APL as a way to feed domain-specific information into the search.

In Gao et al. (2018), a Multi-Objective Evolution Strategy based on CHARMM, is presented. The authors added SASA as one of the three objectives being optimized and bond and non-bond terms from the CHARMM22 energy function. A method using PSO and SVD is presented in Álvarez et al. (2018), where SVD is utilized to

reduce the number of variables, similar to the more common PCA method. Another Multi-Objective PSO is proposed in Song et al. (2018), where it uses two solutions achieves, density estimation and a mutation operator. The system output consists of a set of cluster centroids, which allows the end-user to choose from a set of proteins, the one with the desired properties.

In Narloch and Dorn (2019), the authors present the application of SaDE using data from APL to hybridize the method. The work compared the proposal results with four variants of DE, each with a different operator. The modeling consists of a custom variant of Rosetta's `score3` function using a centroid protein model. A method presented by Varela and Santos (2019) interleaves the stages of Rosetta Classic Abinitio protocol with DE using Crowding. The proposal is to develop a hybrid method that improves upon the Classic *ab initio* protocol from Rosetta by adding several runs of DE.

Another multi-objective EA is presented in Zaman and Shehu (2019), which consists of a memetic EA using MC-based fragment insertion. A crowding operator is utilized to keep the genotypic diversity and prevent some regions of the search space from being over-sampled. The population is generated using the first two steps from Rosetta's Classic *ab initio* protocol.

Based on the literature review, there are several noteworthy points:

- The use of online parameter control (self-adaptive methods) for the PSPP. Only two works (both from the same authors) using online parameter control were found, and one of them is arguably not a pure *ab initio*, namely (Oliveira et al. 2017). Hence, this is an under-explored area of research.
- The use of hybrid methods for the PSPP. While recently, this has received attention, it still has several points left for exploring. One of these points is the use of fragment insertion as a way of integrating domain-specific operators.
- The use of clustering as a way of systematically outputting more than one solution without simply brute-forcing it. In turn, this allows for more direct use of the optimizer by a third-party, which wants to predict the conformation from a protein. This allows for it to choose from a set of conformation.

These points are reinforced in Table 1. Column EA indicates when the method is based on an Evolutionary Algorithm or not. The majority of the methods are EA-based. Column FI shows the use of fragment insertion in the methods. From the 18 related works, 7 used FI. Column OPC indicates the use of Online Parameter Control. Only three works incorporated some form of parameter control. Column LS corresponds to the use of a Local Search procedure. It is worth noting that mutation/crossover operators from methods such as DE and GA are not considered in this column. However, stand-alone procedures interleaved with another method, e.g., an MC search during a PSO, are considered. Column CC considers the use of Conformational Clustering. Of the items considered, this one was the most scarce with only two occurrences. Column KD indicated the use of Knowledge Domain Operators. Fragment indicates that FI was the main source of the knowledge domain. APL and SPL indicates that APL or SPL was utilized, as presented in Borguesan et al. (2015) and Oliveira et al. (2017). From this table, it worth noting that no work found in the literature uses OPC, LS, and CC. Moreover, it is a point worth exploring.

4 Proposed method

This Section presents the methods for approaching the *ab initio* PSPP. Three points need to be specified for an *ab initio* method (Dorn et al. 2014): The protein computational representation, the energy function, and the conformation sampling procedure.

4.1 Computational Representation

This work aims for a full atom prediction of a given target protein, using a full atom backbone representation. The backbone is manipulated by changing the three dihedral angles (ϕ, ψ, ω). The side-chain is abstracted into a centroid of similar mass and shape, maintaining some of its properties. Therefore, the model utilized consists of a backbone torsion angle representation with side-chain centroids.

For this work, the protein model includes more information than just the atoms and its respective conformation. The model is split into three components, as illustrated in Fig. 1, using the 1PLW polypeptide as an example. A first component is the pose object that stores a given protein's conformation considering all its atoms. This object is responsible for updating the atoms when an angle is alternated. Such changes must be propagated down to the chain considering an arm lever effect. The second component of this model consists of a vector which acts as an interface for the optimization algorithm and holds all the backbone dihedral angles. The optimizer does not act directly upon the pose object. Instead, it operates on this vector. When this vector is changed, the pose object changes accordingly. The third component is another vector containing a sequence of predicted secondary structures and its confidence intervals. This information is used to coordinate the sampling procedure.

Table 1 Comparison of the works found in the literature

Source	EA	FI	OPC	LS	CC	KD
Sudha et al. (2015)	Yes	–	Yes	Yes	–	–
Borguesan et al. (2015)	Yes	–	–	–	–	Apl
Garza-Fabre et al. (2016)	Yes	Yes	–	Yes	–	Fragment
Correa et al. (2016)	Yes	–	–	Yes	–	Apl
Narloch and Parpinelli (2017)	Yes	–	–	–	–	–
Hao and Zhang (2017)	–	Yes	–	–	–	Fragment
Hao et al. (2017)	Yes	–	–	–	Yes	–
de Oliveira et al. (2017)	–	Yes	–	Yes	–	Fragment
Oliveira et al. (2017)	Yes	–	Yes	–	–	Apl/spl
Borguesan et al. (2018)	Yes	–	–	Yes	–	Apl
Silva and Parpinelli (2018)	Yes	Yes	–	–	–	Fragment
Kandathil et al. (2018)	–	Yes	–	Yes	–	Fragment
Correa et al. (2018)	Yes	–	–	Yes	–	Apl
Gao et al. (2018)	Yes	–	–	–	–	–
Álvarez et al. (2018)	Yes	–	–	–	–	–
Song et al. (2018)	Yes	–	–	–	Yes	–
Narloch and Dorn (2019)	Yes	–	Yes	–	–	Apl
Varela and Santos (2019)	Yes	Yes	–	Yes	–	Fragment
Zaman and Shehu (2019)	Yes	Yes	–	Yes	–	Fragment
Proposed method	Yes	Yes	Yes	Yes	Yes	Fragment

A dash indicates that the relevant criteria in the column were not met. The last line categorizes the methods proposed in this work

FI Fragment Insertion, *OPC* Parameter Control, *LS* local search, *CC* Conformation Clustering, *KD* Knowledge Domain

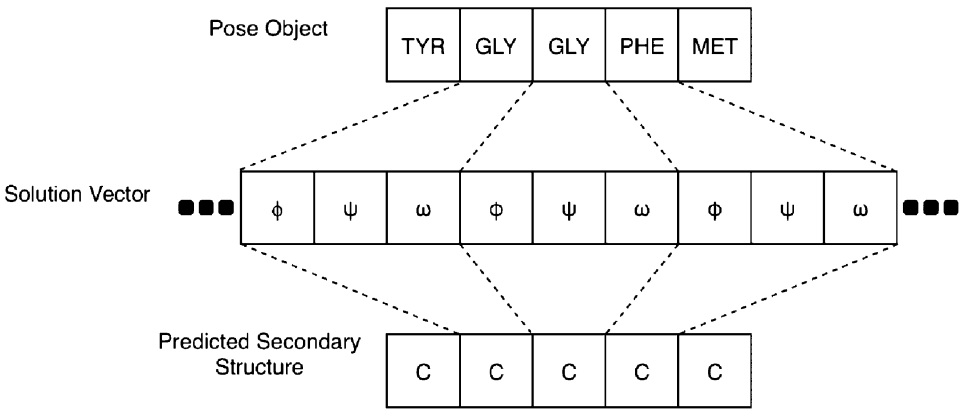

Fig. 1 The protein computational model (Source: Author)

From Fig. 1, the top part consists of the pose model, responsible for holding the atom representation of the protein. The middle part consists of a vector of dihedral angles. Since 1PLW has five amino acids, the angle vector has 15 elements. The first three elements represent the (ϕ, ψ, ω) angles for the first amino acid. The second three elements represent the dihedral angles for the second amino acid, and so on. The bottom part models the predicted secondary structure. For the 1PLW, it consists of only a coil along with all the protein. Each cell of this vector holds the probabilities for each predicted secondary structure. The secondary structures are classified using a DSSP8 notation (Frishman and Argos 1995). For this work they were predicted using the PSIPRED[2] server (McGuffin et al. 2000). Section 4.3 explains how the knowledge from the secondary structure prediction is incorporated in the model using fragment insertion.

At the end of the prediction, the output consists of a full atom protein representation, including the side-chains. For this, an off-the-shelf repacker available in the Rosetta

[2] Available for educational use in: http://bioinf.cs.ucl.ac.uk/psipred/

toolkit was utilized. The repacker replaces the centroids with the actual side-chains. This introduces a series of (potential) clashes between the side-chains, especially in more densely packed structures. A gradient descent optimization is applied to handle these clashes, where the Van der Waals repulsive forces are slightly varied during the gradient descent. The gradient descent can rotate the side-chains and move the backbone to make room for the newly inserted side-chain structures. The final result of this is a full atom representation of the predicted protein, which is the proposed approach's output.

4.2 Energy functions employed

The energy function represents the domain information from the PSPP in a mathematical equation. Different energy functions consider different aspects of the problem, which can be useful in different steps of the optimization algorithm. This work makes use of three energy functions available in Rosetta.

The first energy function utilized is referenced in Rosetta as `score0`. This energy function consists solely of the repulsive Van der Waals forces. The goal of this energy is to aid the generation of initial conformations. Since only the repulsive forces are considered, a `score0` with a value of 0 indicates that a given conformation has no clashes between different parts of the protein. The assembly of the protein guided by the `score0` function leads to more plausible starting proteins.

The second energy function utilized is referenced in Rosetta as `score3`. It uses a full atom representation of the backbone and a centroid to represent the side-chains. A more in-depth explanation of this energy function is explained in Alford et al. (2017). Unfortunately, the only up to date documentation found for the energy functions is the Rosetta source code itself. The `score3` energy function is used during the central portion of the prediction process described next section.

Finally, the last energy function is the `scorefxn`, as found in Rosetta. It encompasses the same information as `score3`. However, it considers a full atom representation of the side-chain as well. This energy function is utilized during the last step of the prediction to output a full atom representation of the protein.

4.3 Conformation sampling

Many metaheuristics can be employed on the PSPP, as shown in Sect. 3. This work employs an Evolutionary Algorithm due to its capability to easily integrate with the necessary tools while detecting promising regions in the search landscape of the problem. Furthermore, an online parameter control technique is used based on the SaDE algorithm (Qin and Suganthan 2005; Qin et al. 2009).

In Kim et al. (2009) is stated that the bottleneck of solving the PSPP is the conformation sampling procedure. Therefore, it is the part that must be more focused on since increasing its performance will likely lead to better predictions. Furthermore, a blind optimizer that does not know the problem domain can inherently have worse performance since it will spend more time sampling regions of the conformation space that are not biologically plausible. Thus, having an efficient sampling procedure and utilizing problem domain knowledge is essential and can lead to a better predictor.

This work makes use of domain-specific operators to integrate the knowledge domain into the optimization process. The domain-specific operator used is fragment insertion. Four fragment insertion operators are employed. Two classic fragment insertion operators of size 3 and 9 are utilized, and two smooth fragment insertions of size 3 and 9.

The classic fragment insertion can be considered a global search operator, as it leads very often to very impactful changes in the conformation. Due to these changes' proportions, this operator will have a decreasing chance of improving the current solution as the optimization process progresses.

Conversely, the smooth fragment insertion can be thought of as being a local search operator. This operator's nature is to try to negate the changes of the first fragment insertion by using a second one. This leads to smaller changes in the overall protein conformation, even though twice the residues are changed. This operator's use allows for the proposed method to have a domain-specific local search, which stays useful throughout the optimization process by operating with small changes.

These four fragment insertion operators require a criterion to be used. For this, an MC based search is employed. The search consists of a series of random fragments insertion, where each fragment insertion is accepted under the MC criterion. A temperature parameter (Cr) determines the likelihood of a degrading (energy-wise) sample being accepted. The fragments that improve upon the energy score are always accepted, while fragments that deteriorate the energy score are more likely to be kept based on how little it affected the energy. This allows for the search to potentially escape from being trapped in minima regions. Furthermore, it helps to navigate through the rugged energy landscape of the protein potential energy function.

Another essential tool in the conformation sampling step is the use of Forced Fragment Insertion (FFI). FFI consists of inserting a random fragment regardless of its impact. Unlike the MC fragment insertion, FFI applies the fragment without considering the energy impact it has.

Determining when FFI occurs is of paramount importance. If it happens too often, suitable conformations will keep being destroyed, and the sampling procedure will be impaired. If it seldom happens, then the benefit of escaping local minima will rarely be used. Therefore, the strategy has to be tuned so that we escape local minima often enough to avoid wasting too many function evaluations while stuck but not so often so that too many suitable conformations are destroyed. Also, exploring local minima is by itself import. While the optimization is happening, there is no way of knowing if the best local minima found so far is the global minima or not. If local minima are not explored enough, FFI may prevent the optimal point from being found.

Despite its potential downsides, using FFI can help prevent premature convergence in the system. It adds diversity to the conformation pool in a controlled manner. With that, a constant stream of information is added to the optimizer. Finally, coupled with the optimizer itself, FFI acts as a catalyst for the optimizer to make significant conformation changes because FFI can bypass the EA's greedy nature.

Figure 2 presents a flowchart of the proposed approach. It shows the initialization phase, the optimization phase, and the post-processing phase.

In the initialization phase, the primary sequence is taken as input in the FASTA format, consisting of the one-letter code sequence of amino acids. The primary sequence is stored for later use and feeds the PSIPRED secondary structure predictor. PSIPRED outputs a probability matrix mapping probabilities for the secondary structures' cartesian product versus the amino acid sequence. This output is also stored for later use and feeds the fragment picker. The fragment picker used is the Rosetta Fragment Picker. The fragment picker is responsible for selecting a set of fragments for all possible combinations of contiguous amino acids of size 3 and 9. The fragment set is stored to be used as input for the tertiary prediction routine. Since the initialization phase consists only of preprocessing, its time is not considered when measuring the time required to predict a given target protein, especially considering the waiting time for the PSIPRED server.

The second phase is the optimization phase. It starts with the step of generating the initial population. The initial population generation consists of assembly random protein conformations using the fragments generated in the initialization phase. A Monte Carlo search is run using the `score0` energy function to search for protein conformations with no (or as few as possible) hysterical clashes. This step stops when a fixed number of samples is used for each solution vector in the population or when the `score0` function reaches zero.

With the initial population generated, the optimization procedure itself starts guided by the `score3` energy function. The self-adaptive evolutionary algorithm carries out the basis of the search procedure. The hybridization happens when the *Domain Operators* are added to the search procedure.

Firstly, the proposed method's online parameter control portion selects which operator to use for each individual in the population.

The operators consists of a MC fragment insertion with fragments of size either 3 or 9 using a *smooth* or a *classical* fragment insertion. Once it is selected, a small sub search procedure starts, corresponding to the operator itself. Before the operator is applied, the check for FFI happens. The check occurs for all solution vectors. If the FFI check passes, then a random fragment is applied. Regardless of FFI being used, the next step is to apply the probabilistic fragment insertion operator. An MC search procedure is initialized based on the operator chosen. This search takes a fixed amount of function evaluation that is fed as a parameter. When this search procedure stops, the output is fed to the greedy selection routine. Each solution vector in the population is compared to its respective trial vector. Greedy selection is performed in which the trial vector is accepted if and only if it has a better `score3` value than its current solution vector. From there, the stop criteria is verified. If it is not met, then the parameter update routine is called, updating the parameter Cr (for all operators) and the probabilities for each operator. The *Cr* parameter represents the MC temperature parameter. The parameter control is agnostic to which parameter it is updating. Once the Parameter Update step is finished, the optimization cycle starts over again from the operator selection.

When stop criteria are met, the optimization phase stops, and the post-processing phase starts. The first post-processing step is to cluster the conformations. After the clustering process finishes, the cluster centroids are selected. The number of clusters is a parameter of the algorithm. The conformations closest to the centroids are found and feed-forward to the next steps. A Hooke-Jeeves local search is applied to acting on all three dihedral angles of the full protein backbone. This helps to reach nearby local minima that might have been inaccessible by the fragments alone.

Once the local search finishes, the repacking procedure is applied to all conformations that were selected in the clustering phase and then re-optimized. The repacking procedure removes the centroids and places rotamers (side-chains fragments). The conformation is again re-optimized, this time using a gradient descent guided by the `score_fxn` energy function. After this step finishes, the conformations are returned.

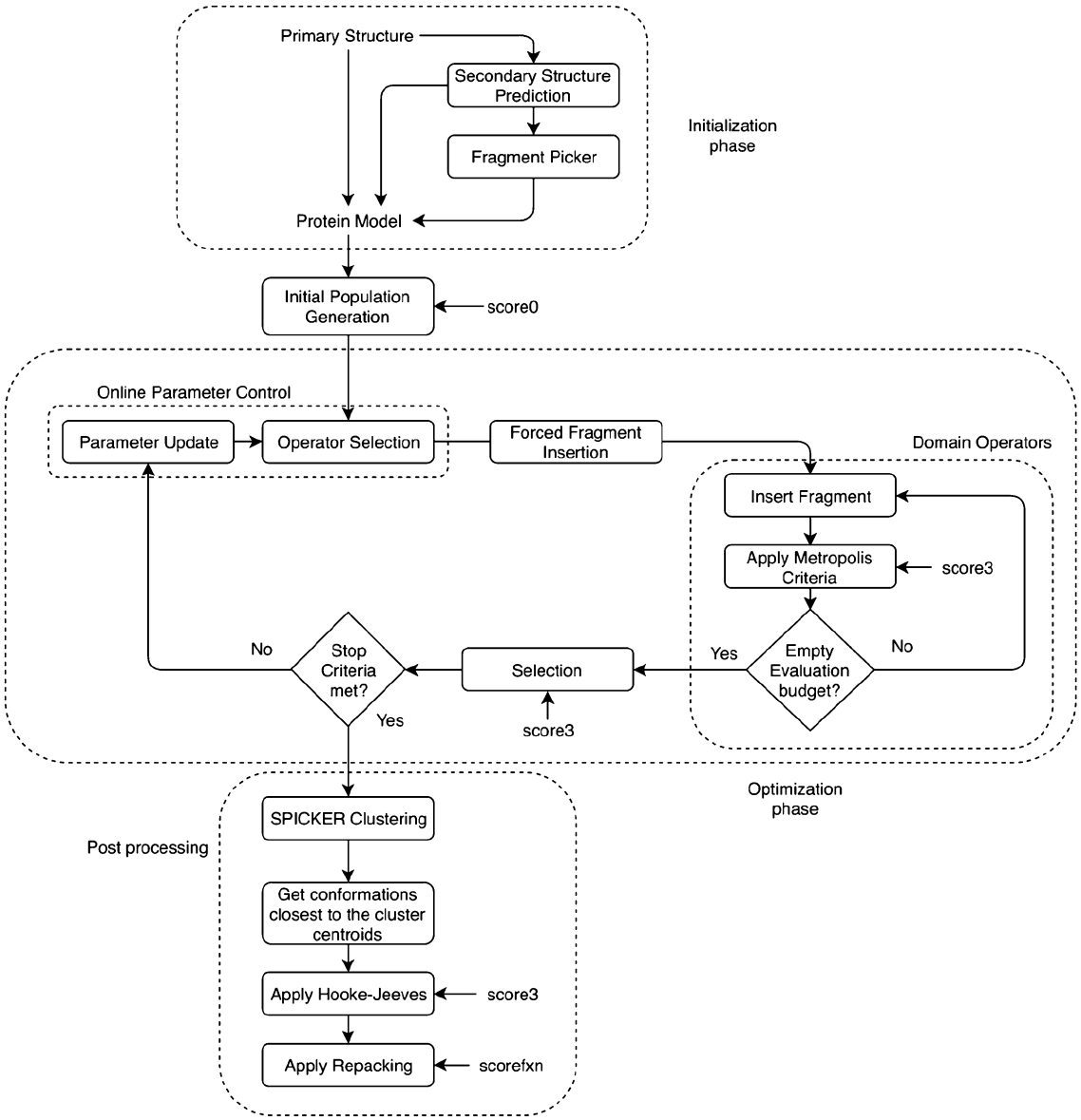

Fig. 2 The Proposed search procedure (Source: Author)

5 Experiments, results and analysis

This section presents the design of experiments, the energy and RMSD analysis, the processing time, the comparison with competing methods in the literature, the GDT-TS, TM-Score analysis, and the predicted conformations' visual analysis.

5.1 Design of experiments

The experiments were all conducted on a single machine using the same hardware throughout the full experimentation. Table 2 presents the machine utilized to run all the experiments. Each run of a prediction method consists of a serial program that runs continuously without interruption. The experiments were run in parallel, limited to at most

Table 2 The Machine Setup

Name	Value
Operating system	Arch Linux
Kernel	Arch Linux Kernel 4.18.16
CPU	Intel(R) Core(TM) i5-3570K CPU @ 4.20GHz
Number of cores	4 Physical cores, no hyper-threading cores
RAM	16 GB @ 1400 MHz

one running test per core[3]. To ensure maximum repeatability, the machine had no graphical interface enabled or any other user interaction form during the experimentation. The proposed method was developed using the Python language. The experimentation consisted of running two methods: The proposed method, namely PPF-MC[4], and the Rosetta Ab Initio protocol.

The metrics utilized are the *scorefxn* energy value of the best solution and the RMSD associated with the conformation. The results were collected over 50 independent runs of each method for each target protein. A rigorous numerical statistic set of tests is conducted. The Shapiro-Wilk (Wilk and Shapiro 1968) normality test is employed with a confidence level of 5%, i.e., $\alpha = 0.05$, to assess the presence (or lack) of an underlying normal distribution. Based on its result, a parametric/non-parametric test is employed with a confidence level of $\alpha = 0.05$. Due to the presence of multiple comparisons, the Kruskal-Wallis test is applied to detect any method with different performances. Then, the pairwise Mann-Whitney test is employed with the proposed method against its competitors. Also, graphical analysis is conducted in order to identify the relative performance of the proposed method visually.

Clustering to extract and return different conformations from the proposed method is essential in a complex and extremely multi-modal problem such as the PSPP. With clustering, it is possible to identify conformations that are far apart from each other in the energy landscape but have similar energies. This process requires extra steps during the analysis. First, the primary use of returning several conformations is to allow a human expert to choose one with the desired properties. As such, the human expert must be replaced by a computer oracle for performance evaluation. This oracle can always find the conformation with the lowest RMSD or the conformation with the lowest energy. Of course, this would not be possible in a real-world scenario where a protein without a known structure is predicted. With that in mind, the proposed method's analysis is based on the best conformation, as measured by RMSD.

Also, a direct comparison against several works in the literature is considered. Since there is a severe lack of standardization in the literature regarding experimentation, the following methodology was used. Works that provided the best RMSD had their proteins listed. The proteins that occurred the most were used for comparison. It is worth noting that the majority of works provide little information about how the experimentation was conducted. As such, this work does a direct comparison using the best RMSD achieved in a set of runs. While this is not ideal, due to different works running different methods, this is possibly the only way to compare several works. Nevertheless, at the end of the day for the PSPP, what matters is having the lowest possible error. Also, comparing the best RMSD is a worthwhile analysis.

With that in mind, Table 3 presents the set of chosen proteins. The column *Name* contains the protein identification code as in PDB. The *Size* column shows the number of amino acids in the protein. The *Backbone Angles* column shows the number of angles in the backbone. The *Structure* column holds the secondary structures present in the protein set represented by α-helices or β-sheets.

The proposed method operates with the parameters presented in Table 4. The first column contains the parameter name, and the second column presents its respective value. The Population Initialization column refers to the MC search that is made using `score0`. It has 10, 000 function evaluations available, such that up to 100 are used for each solution vector. The self-adaptive learning phase has its default value, as presented in Qin et al. (2009). There are 100 simultaneous trajectories throughout the execution, i.e., a population size of 100. A million function evaluations are available for the optimization phase, where each fragment insertion routine can use up to 25 at a time. FFI uses a fragment size of 9 and is applied with a probability of 2% before each standard fragment insertion. The other methods being compared use the same values from Table 4 as applicable. The function evaluation budget available for the Rosetta Ab Initio protocol is the same as the proposed method. However, the Rosetta Suite has some specific stopping criteria in its definitions that can make the optimization process stop before using all function evaluations available.

Table 3 Target proteins and their features

Name	Size	Backbone angles	Structure
1l2y	20	60	2α
1wqc	26	78	2α
1acw	29	87	$1\alpha, 2\beta$
1zdd	35	105	2α
2mr9	44	132	3α
1crn	46	138	$2\alpha, 2\beta$
1enh	54	162	3α
1rop	63	189	2α
1utg	70	210	4α
1ail	72	216	3α

[3] Only physical cores were considered. No virtual (Hyper-threading) core was involved in the computations.

[4] Source code available at https://github.com/h3nnn4n/protein-prediction-framework/

Table 4 Parameters utilized in the proposed method

Parameter	Value
Population initialization evaluation budget	10,000
Learning phase	50
Population size	100
Function evaluation budget	1,000,000
MC function evaluation budget	25
Spicker cluster size	10
FFI probability	0.02
FFI length	9

5.2 Energy and RMSD analysis

The analysis is divided into two parts. In the first part, a visual analysis using box-plots is used to observe the proposed method's overall performance compared with the classical *ab initio* method provided by the Rosetta Suite. In the second part, a statistical framework is used to assess the methods' performance relative to each other. All analyses are based on the results collected over 50 independent runs of each method for each target protein.

In Fig. 3, the RMSD from the predictions is presented as box-plots. The proteins, presented on the x-axis, are displayed in lexicographical order. The y-axis presents the RMSD, where a lower value is most desirable. The methods are grouped horizontally by protein.

In a direct comparison against Rosetta, proteins 1acw, 1enh, 1l2y, 1rop, 1utg, and 2mr9, the proposed method had significant improvements. For 1crn, the Rosetta appears to have had slightly better performance compared to the proposed method. In the remaining proteins, 1ail, 1wqc, and 1zdd, it is not possible to visually detect objectively if a significant improvement is present. This will be addressed later with proper statistical tests.

Figure 4 presents data similarly to the previous figure, however, the y-axis now represents the `scorefxn` energy function. Considering the energy results, when compared to the RMSD boxplot, the results are relatively more similar. The main differences appear for 1utg and 2mr9, where rosetta appears to be lagging in performance. For 1ail, 1enh and 1zdd rosetta had a considerably bigger variance than the proposed method. These observations are just visual trends which help understand the relative performance.

For the statistical analysis of the results obtained, the Mann-Whitney test was applied with $\alpha = 0.05$. The null hypothesis is that the two distributions are equal, i.e., both methods have the same performance. Rejection of H_0 indicates that one method is better than the other. Considering that both RMSD and `scorefxn` will be analyzed, each for ten proteins, a total of 20 tables are necessary to expose all the data. As such, this information is not exposed in this work. Instead, the results are summarized, reporting the overall results from the test. The proposed method is compared to rosetta, considering both RMSD and `scorefxn` in the analysis.

The results of Mann-Whitney's statistical tests for the RMSD are shown in Table 5. The first column indicates the protein name. The second, third, and fourth columns indicate if PPF-MC, Rosetta, or neither had a statistically significant performance difference. The proposed method, PPF-MC, had a statistically significant difference measured in the RMSD for proteins 1acw, 1enh, 1l2y, 1rop, 1utg, and 2mr9. The only occasion where Rosetta outperformed the proposed method was on 1crn. On proteins 1ail, 1wqc, and 1zdd, the performance was statistically equivalent. This matches the previous observations from Fig. 3. The proposed method had a superior performance for six proteins, a performance worse than Rosetta on one protein, and an equivalent performance in three proteins.

The potential energy results from Mann-Whitney's test, as measured by the `scorefxn` function, is presented in Table 6. Its interpretation is the same as Table 5. Interestingly, for proteins 1acw, 1l2y, and 1rop PPF-MC had a statistically equivalent performance when measured by the potential energy, while it had a statistically superior performance when measured by the RMSD. This indicates that while both methods found regions of similar potential energy levels, the conformations from PPF-MC had a lower RMSD for the same energy levels. Conversely, the same can be said for Rosetta on 1crn. On 1ail, PPF-MC had a statistically equivalent performance measured by the RMSD. However, when measured by potential energy, the performance was superior, as indicated by the test results. The proposed method had a superior performance on four proteins, considering the potential energy: 1ail, 1enh, 1utg, and 2mr9. Rosetta outperformed PPF-MC in no proteins when considering the potential energy. Both methods had an equivalent performance in six proteins: 1acw, 1crn, 1l2y, 1rop, 1wqc, and 1zdd.

On three proteins, 1enh, 1utg, and 2mr9, PPF-MC had a superior performance than Rosetta on energy and RMSD. Considering the 20 scenarios analyzed, from 10 proteins $\times$ 2 metrics (RMSD and Energy), in 10 situations, PPF-MC had superior performance. Rosetta had a superior performance in only one scenario: the RMSD on protein 1crn. In the other nine scenarios, both methods had a statistically equivalent performance. Overall, the proposed method had a performance equivalent or better than Rosetta in 19 of the 20 scenarios.

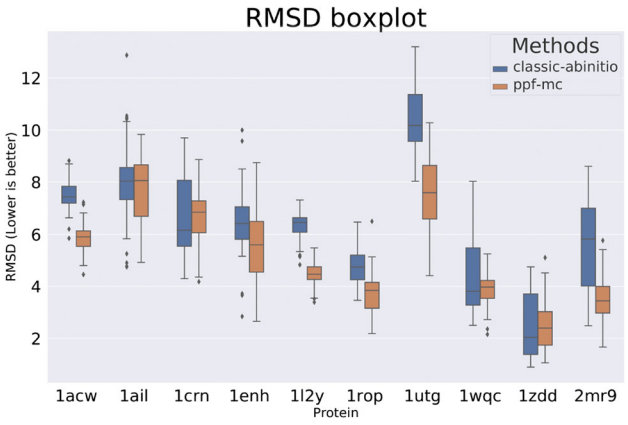

Fig. 3 Box-plot presenting the RMSD obtained by PPF-MC and Rosetta

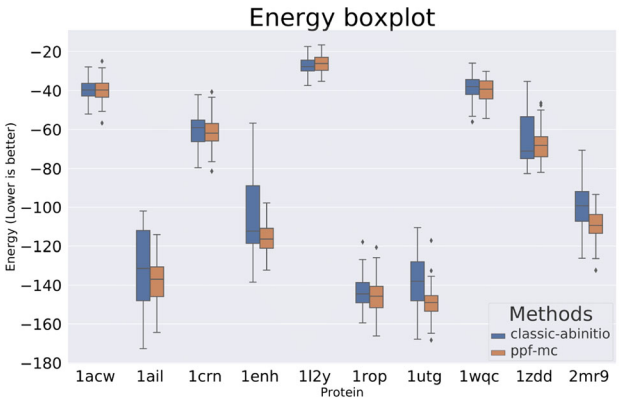

Fig. 4 Box-plot presenting the `scorefxn` for the protein predictions obtained by PPF-MC and Rosetta

Table 5 Mann-Whitney results applied to RMSD

Protein	PPF-MC	Rosetta	Draw
1acw	x		
1ail			x
1crn		x	
1enh	x		
1l2y	x		
1rop	x		
1utg	x		
1wqc			x
1zdd			x
2mr9	x		
Total	6	1	3

Table 6 Mann-Whitney results applied to Energy

Protein	PPF-MC	Rosetta	Draw
1acw			x
1ail	x		
1crn			x
1enh	x		
1l2y			x
1rop			x
1utg	x		
1wqc			x
1zdd			x
2mr9	x		
Total	4	0	6

Table 7 Processing time, in seconds, for PPF-MC and Rosetta Suite

Protein	PPF-MC	Rosetta
1acw	381.8566 ± 76.8664	105.3112 ± 16.5700
1ail	1141.1461 ± 142.2935	253.3252 ± 48.9487
1crn	721.0492 ± 118.6126	283.6198 ± 51.6943
1enh	763.7677 ± 86.2599	208.056 ± 44.3525
1l2y	252.8980 ± 19.1262	109.258 ± 8.8145
1rop	963.2695 ± 124.7610	158.6072 ± 16.3965
1utg	1010.0412 ± 142.8927	184.7794 ± 8.9843
1wqc	341.4484 ± 35.1510	71.4528 ±0.7017
1zdd	462.3010 ± 127.2581	94.2868 ± 3.8235
2mr9	702.5817 ± 95.6492	138.6254 ± 3.5222

5.3 Processing time and function evaluations

Table 7 shows the processing time obtained in the experiments reported in seconds. The first column shows the protein name. The second column shows the mean, and the third column shows the standard deviation. The total processing time does not include the preprocessing steps. The time starts counting for the main optimization phase when the initial population is generated.

As expected, there is a direct correlation between the number of residues in a given protein and the time required to predict its structure. The two smallest proteins, 1l2y and 1wqc, had faster processing times, while larger proteins, 1utg and 1ail, had the highest times. The prediction times, on average, range from about 4 minutes up to 20 minutes per run. As observed, the Rosetta Ab Initio protocol (classical method) is faster than the proposed approach. This can be explained because the PPF-MC is a population-based approach (multi-trajectory method) and the classical method is a single-trajectory method.

Another performance analysis that can be made in studying the spent function evaluations during the local search phase. The main optimization phase has a fixed

budget of 1 million function evaluations. The post-processing phase, however, uses a separate budget, which is non-fixed. The Hooke-Jeeves search procedure is applied multiple times, successively until no improvement is detected. If a single call to Hooke-Jeeves spends more than 5000 evaluations, it is flagged for termination at the next iteration. As such, each call spends around 5000 to 6000 evaluations. However, multiple calls can be made in succession.

Table 8 presents the mean spent function evaluations and the respective standard deviations. For all ten proteins, the mean stays relatively close to 20,000. A manual inspection of the logs reveals that there is no spent more than 100,000 evaluations.

5.4 Comparison with competing methods

A comparison against methods in the literature is challenging to conduct. Most of the literature's methods are relatively superficial in explaining how a given algorithm was implemented and the employed testing methodology. Paper space seems to be a possible cause for this since articles that span more pages are usually more detailed about the implementation and methodology. As such, a comparison has to be based on the data provided in the works, which in most cases, is not enough for a proper rigorous analysis. Nevertheless, this work attempts to provide a simple framework for comparing the proposed method with works in the literature. Several works were selected, where the model utilized was the full atomic model with an *ab initio* method. Their proteins and the RMSD of the best prediction was recorded.

In Table 9, the first column indicates the year of the publication presented in decreasing order. The column *Source* presents the source of the data, which is the proposed method, Rosetta, or work from the literature. The remaining columns present several proteins, sorted by the frequency in which it appears in the literature. The protein 1rop is the most frequent protein, while 1wqc is ranking 10th in the frequency list. The data in these columns is the best RMSD from the method in the given work. Overall best results are highlighted in bold. The comparison takes into account works found between 2018 and 2019.

The proposed method was able to achieve the best RMSD for several proteins. For proteins 1rop, 1crn, 1enh, 2mr9, 3.39, 1ail, and 1wqc, the proposed method had the best RMSD considering the past two years. On 1utg, the proposed method had the second-best RMSD, with a difference of 0.12Å to a result from literature. For 1zdd, the proposed method was ranked second, with an RMSD difference of 0.16Å, with Rosetta ranked first. On 1acw, the proposed method ranked third, with a difference of 2.78Å. In light of this, it is reasonable to consider PPF-MC as a strong competitor of the state-of-the-art methods.

Another point worth stating is that some proteins might be way too easy for the current methods. Take 1zdd, for example, had RMSD smaller than 2.62. Considering that most PDB proteins have a resolution ranging from 1 to 2Å, trying to go smaller than that is more a pursue of luck than science. As such, this protein might only be useful for validating new methods, but not for measuring progress.

Table 8 Function evaluations spent on Hooke-Jeeves, for PPF-MC

Protein	Mean	stddev
1acw	19894.6690	9323.3251
1ail	23215.9890	12096.6411
1crn	23704.9947	11297.4331
1enh	21141.6667	11922.8046
1l2y	16022.7659	7145.3393
1rop	21008.9153	11595.2773
1utg	23257.8303	13708.3805
1wqc	18342.0987	8163.8972
1zdd	21848.3618	10229.3850
2mr9	22260.8176	11415.6200

5.5 GDT-TS and TM-Score metrics

This section provides an in-depth analysis of the results obtained using the GDT-TS and the TM-Score metrics. The GDT-TS and TM-Score measurements are used as major assessment criteria in the production of results from the Critical Assessment of protein Structure Prediction (CASP[5]). Also, they are intended as a more accurate measurement than the more common RMSD metric (Zemla 2003). The conformations analyzed are the ones that had the best RMSD. The results can also be used by other works to compare against our own using these metrics. Tables 10 and 11 present the GDT-TS and TM-Score values, respectively. Both tables follow the same format. The first column presents the protein name. The results are presented in separate columns with the best result, the mean, and the standard deviation.

Both GDT-TS and TM-Score share the same property where values can be used as thresholds for prediction quality. A value close to 0.2 indicates a random prediction performance, while a value of 0.5 or above suggests a prediction that has the same overall fold. Considering that, values of 0.5 or above are marked in boldface font. Values closer to 1.0 indicates a near-perfect prediction.

[5] CASP website: https://predictioncenter.org/

Table 9 A comparison of the RMSD from the best prediction

Year	Source	1ROP	1CRN	1UTG	1ZDD	1ENH	2MR9	1L2Y	1ACW	1AIL	1WQC
	ppf-mc	**2.18**	**4.18**	4.41	1.07	**2.65**	**1.66**	**3.39**	4.45	**4.26**	**2.15**
	Rosetta	3.46	4.30	8.03	**0.91**	2.84	2.48	4.83	5.85	4.75	2.50
2019	Silva and Parpinelli (2019)	–	6.08	–	1.16	3.23	–	–	–	4.46	–
2019	Narloch and Dorn (2019)	6.02	4.53	6.38	2.35	5.56	2.49	–	**1.67**	–	–
2018	Song et al. (2018)	2.21	5.16	5.68	1.84	5.81	–	–	–	–	–
2018	Borguesan et al. (2018)	–	–	**4.29**	–	–	2.39	–	2.00	–	–
2018	Silva and Parpinelli (2018)	–	6.96	–	2.62	5.70	–	–	–	8.27	–

Table 10 GDT-TS for PPF-MC

Protein	Best	Mean	Stddev
1acw	***0.5172***	0.4605	0.0302
1ail	0.4932	0.3794	0.0473
1crn	***0.5870***	0.4343	0.0560
1enh	0.4861	0.4262	0.0369
1l2y	***0.6625***	***0.5773***	0.0439
1rop	***0.6825***	***0.5772***	0.0517
1utg	***0.5571***	0.4346	0.0631
1wqc	***0.7885***	***0.6446***	0.0451
1zdd	0.4412	0.4193	0.0160
2mr9	***0.8807***	***0.6891***	0.0720

Table 11 TM Score for PPF-MC

Protein	Best	Mean	Stddev
1acw	0.2475	0.1930	0.0202
1ail	0.4468	0.3039	0.0457
1crn	0.3986	0.2762	0.0462
1enh	0.3489	0.2628	0.0272
1l2y	0.2495	0.1924	0.0294
1rop	***0.6229***	0.4588	0.0646
1utg	0.4938	0.3676	0.0632
1wqc	0.3757	0.2852	0.0346
1zdd	0.3178	0.2797	0.0276
2mr9	***0.7514***	***0.5117***	0.0900

From Table 10, GDT-TS values, for 1zdd, the protein with the lowest RMSD has a GDT-TS of 0.41. Meanwhile, 1wqc and 2mr9 had values above 0.75 and 0.87, respectively, which means that the two metrics, namely GDT-TS and RMSD, do not always agree.

Considering the TM-Score, there are a few numbers of values above the threshold of 0.5. Furthermore, RMSD and GDT-TS disagree on some cases, such as for 1zdd, 1wqc, and 2mr9, but TM-Score can differ from the other two metrics in some cases as well. For instance, on 1wqc, the GDT-TS value was 0.75 or higher, while the respective TM-Scores were 0.31 or lower.

One noteworthy aspect of GDT-TS and TM-Score is that they appear to be more rigorous than RMSD. For example, the 1enh protein has the second-best RMSD found in the literature, yet, with GDT-TS, it did not cut 0.5. More so, both 1enh and 1ail, two relatively large proteins, had their best RMSD more than two units apart with PPF-MC. However, with the GDT-TS, the two conformations are less than 0.01 units apart. This is possible due to RMSD being a metric with an unbounded upper limit, which scales quadratically with the number of residues. GDT-TS, on the other hand, has a normalized value between 0 and 1, which allows the predictive performance to be compared not only across different methods but across proteins of different sizes. Unfortunately, very few works in the literature use these metrics.

Furthermore, considering that TM-Score tended to underestimate the predictions' quality, it might not be the best for tracking performance in a method under development. On the other hand, GDT-TS was able to identify both good and bad predictions, making it a more suitable metric for such conditions.

5.6 Visual representation of the predictions

Figure 5 all conformations from the ten best predictions measured by RMSD from PPF-MC are presented in lexicographical order. The predicted conformation is presented in green, and the native conformation is presented in red[6].

Proteins 1enh, 1rop, 1utg, 1wqc, 1zdd, and 2mr9 had near-native conformations. All the secondary structures are present in their respective regions, and the coil sections

[6] For the readers with a black and white copy, the predicted conformation is in a light shade of grey, while the native conformation is in dark grey.

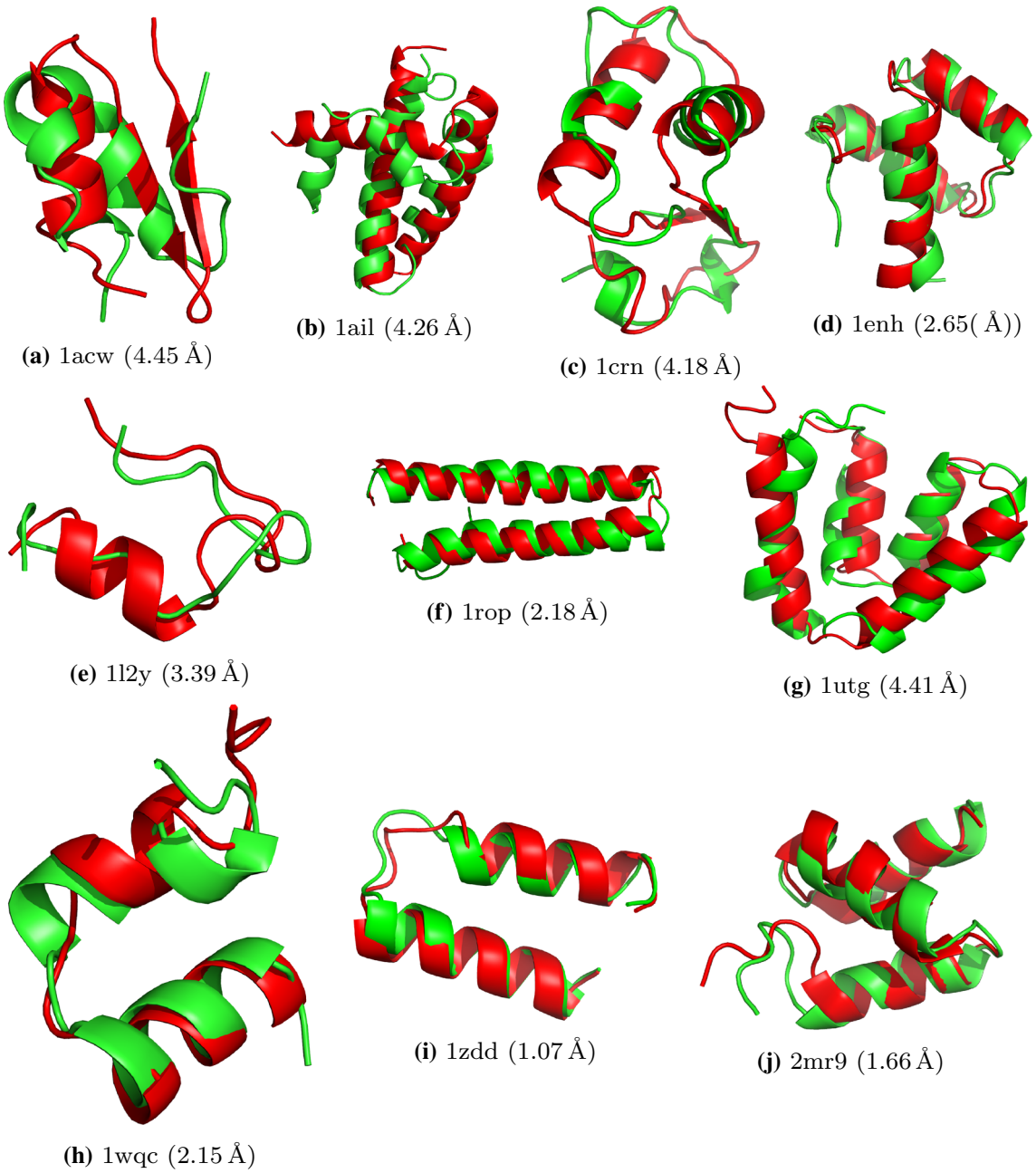

Fig. 5 The predicted conformations (in green/light gray) compared to the native conformation (in red/darker gray). The RMSD between the predicted and native conformation is show between parenthesis

closely match their native counterparts. For protein 1acw, which had a relatively high RMSD, considering the size of the protein, there are two main prediction errors. Firstly, the β-sheets did not form, and where should be one, there is an α helix instead. The second error is that the α helix is in the wrong place and split. It starts where it should, but it only has a single turn. A second helix forms at the eighth residue and goes on for ten more residues. Both these errors can be traced down to an error in the Secondary Structure Prediction, which the proposed method has no way to avoid. For 1ail, the prediction is mostly correct. However, two helices are split apart. The first, to the left of the image, and the second, in the middle helix. These errors were prediction failures that occurred in the proposed method.

On 1crn, a relatively complex protein has an overall correct fold. The fine details, however, are lacking. The two helices are mostly missing, and the sheets did not fold. These two errors can be traced down to the secondary structure prediction used as input. For 1l2y, a similar scenario occurs, where the primary source of error is in the

data fed to the prediction engine. The overall shape of the protein was correctly predicted. The helix is entirely missing.

Interestingly, on 4 of the proteins with the biggest visually detectable error (1acw, 1ail, 1crn, and 1l2y), 3 of them (1acw, 1crn, and 1l2y) had its source of error outside to the prediction. The proposed method relies heavily on the predicted secondary structure and has to deal with uncertainty. In only a single case, the significant error in a prediction was generated during the prediction itself, as occurred on 1ail.

6 Conclusions and future works

This work proposed the PPF-MC algorithm to attempt to solve the Protein Structure Prediction Problem (PSPP). The protein is modeled using a full atomic model of the backbone, manipulated using fragments and torsion angles, and side-chain centroids. The proposed method uses the score0, score3, and scorefxn energy functions from the Rosetta suit. Also, a clustering process to generate several conformations is employed.

The proposed method was compared against Rosetta using both the RMSD and potential energy. For both metrics, a rigorous set of statistical tests was applied, which identified that the proposed method could give equal or better predictions than Rosetta for the majority of proteins.

Comparing the results obtained from the PPF-MC with the results obtained from state-of-art algorithms, highly competitive achievements are reported. For two target proteins 1wqc and 2mr9, the proposed method was able to overpass the best result found in the literature. For the 1rop and 1crn proteins, the PPF-MC could get the second-best prediction. The same occurred for 1enh protein. The proposed method achieved results that were often close to the best results for the remaining target proteins.

An analysis using GDT-TS and TMScore metrics demonstrated that PPF-MC appears to be more strict about the results than RMSD. Also, results obtained showed that the three metrics do not always agree with values significantly apart. This study gave the insights that the prediction could be improved even further.

Visual analysis was performed, where four proteins were detected as having significant errors, and three of them had its errors tracked down to the predicted secondary structure, which is the input of the proposed method. We can conclude that this dependency on predicted information is potentially one of the proposed method's significant weak-spots. Therefore, a future research direction would be to employ more than one predictor or research to make the tertiary structure prediction either less dependent on the secondary structure prediction or make it more resilient to predicted information.

The use of other fragments generator is also a research direction since different fragment libraries may lead to different predictions. Another possible direction is the inclusion of gradient descent methods to quickly find nearby low-energy areas because acting on population diversity might further extend the potential of using conformational clustering and lead to better predictions. Another way of acting on population diversity is to periodically reset parts of the population or the whole population if stagnation occurs. In particular, partial population re-initialization can allow for old information to contribute to the newer generated individuals. Still about diversity, one way would be to implement some speciation or subpopulation technique, as in Deng et al. (2020) and Tawhid and Ali (2017), respectively.

The proposed method applied the Hooke-Jeeves pattern search as one of the last steps. However, one direction would be to use the local search procedure as an operator for the evolutionary algorithm. Hence, allowing for the method to quickly find local regions of good potential energy.

Another recent trend worth exploring is focusing the search on loops and coil regions of the proteins. These areas usually have a lower acceptance rate of perturbation due to the high impact on the overall conformation. In contrast, α-helices are relatively stable and have a much higher acceptance rate while having a low impact on the overall structure.

Also, performing a factorial experiment concerning the proposed approach's components may better understand their impact on the results.

References

Alford RF, Leaver-Fay A, Jeliazkov JR, O'Meara MJ, DiMaio FP, Park H, Shapovalov MV, Renfrew PD, Mulligan VK, Kappel K et al (2017) The Rosetta all-atom energy function for macromolecular modeling and design. J Chem Theory Comput 13(6):3031–3048

Álvarez Ó, Fernández-Martínez JL, Cernea A, Fernández-Muñiz Z, Kloczkowski A (2018) Protein tertiary structure prediction via svd and pso sampling. In: International conference on bioinformatics and biomedical engineering. Springer, pp 211–220

Anfinsen CB (1973) Principles that govern the folding of protein chains. Science 181(4096):223–230

Berger B, Leighton T (1998) Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. J Comput Biol 5(1):27–40

Blum C, Puchinger J, Raidl GR, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. Appl Soft Comput 11(6):4135–4151

Boiani M, Parpinelli RS (2020) A GPU-based hybrid jDE algorithm applied to the 3D-AB protein structure prediction. Swarm Evol Comput 58:100711

Borguesan B, e Silva MB, Grisci B, Inostroza-Ponta M, Dorn M (2015) Apl: an angle probability list to improve knowledge-based metaheuristics for the three-dimensional protein structure prediction. Comput Biol Chem 59:142–157

Borguesan B, Narloch PH, Inostroza-Ponta M, Dorn M (2018) A genetic algorithm based on restricted tournament selection for the 3d-psp problem. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE, IEEE, Rio de Janeiro, Brazil, pp 1–8. https://doi.org/10.1109/CEC.2018.8477721

Brooks BR, Brooks CL III, Mackerell AD Jr, Nilsson L, Petrella RJ, Roux B, Won Y, Archontis G, Bartels C, Boresch S et al (2009) Charmm: the biomolecular simulation program. J Comput Chem 30(10):1545–1614

Correa L, Borguesan B, Farfan C, Inostroza-Ponta M, Dorn M (2016) A memetic algorithm for 3-D protein structure prediction problem. IEEE/ACM Trans Comput Biol Bioinf 15:690

Correa LDL, Dorn M (2018) A knowledge-based artificial bee colony algorithm for the 3-d protein structure prediction problem. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE, Rio de Janeiro, Brazil, pp 1–8

David A, Islam S, Tankhilevich E, Sternberg MJ (2022) The AlphaFold database of protein structures: a biologist's guide. J Mol Biol 434(2):167336

de Oliveira SH, Law EC, Shi J, Deane CM (2017) Sequential search leads to faster, more efficient fragment-based de novo protein structure prediction. Bioinformatics 34(7):1132–1140

Deng L, Zhang L, Sun H, Qiao L (2020) Dsm-de: a differential evolution with dynamic speciation-based mutation for single-objective optimization. Memet Comput 12:73–86. https://doi.org/10.1007/s12293-019-00279-0

Dorn M, e Silva MB, Buriol LS, Lamb LC (2014) Three-dimensional protein structure prediction: methods and computational strategies. Comput Biol Chem 53:251–276

Eichenberger AP, Allison JR, Dolenc J, Geerke DP, Horta BA, Meier K, Oostenbrink C, Schmid N, Steiner D, Wang D et al (2011) Gromos++ software for the analysis of biomolecular simulation trajectories. J Chem Theory Comput 7(10):3379–3390

Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. Proteins Struct Funct Bioinf 23(4):566–579

Gao S, Song S, Cheng J, Todo Y, Zhou M (2018) Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction. IEEE/ACM Trans Comput Biol Bioinf 15(4):1365–1378

Garza-Fabre M, Kandathil SM, Handl J, Knowles J, Lovell SC (2016) Generating, maintaining, and exploiting diversity in a memetic algorithm for protein structure prediction. Evol Comput 24(4):577–607

Geng L, Shen H (2017) A protein structure refinement method using bi-objective particle swarm optimization algorithm. Image and Signal Processing. In: BioMedical Engineering and Informatics (CISP-BMEI), 2017 10th international congress on. IEEE, Shanghai, China, pp 1–5

Gront D, Kulp DW, Vernon RM, Strauss CE, Baker D (2011) Generalized fragment picking in Rosetta: design, protocols and applications. PloS One 6(8):e23294

Gunn JR (1997) Sampling protein conformations using segment libraries and a genetic algorithm. J Chem Phys 106(10):4270–4281

Hao X, Zhang G (2017) Double estimation of distribution guided sampling algorithm for de-novo protein structure prediction. In: Control Conference (CCC). 2017 36th Chinese. IEEE, Dalian, China, pp 9853–9858

Hao XH, Zhang GJ, Zhou XG (2017) Conformational space sampling method using multi-subpopulation differential evolution for de novo protein structure prediction. IEEE Trans Nanobiosci 16(7):618–633

Hart WE, Istrail S (1997) Robust proofs of np-hardness for protein folding: general lattices and energy potentials. J Comput Biol 4(1):1–22

Higgs T, Stantic B, Hoque MT, Sattar A (2010) Genetic algorithm feature-based resampling for protein structure prediction. In: Evolutionary computation (CEC). 2010 IEEE congress on. IEEE, Barcelona, Spain, pp 1–8

Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A et al (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596(7873):583–589

Kandathil SM, Garza-Fabre M, Handl J, Lovell SC (2018) Improved fragment-based protein structure prediction by redesign of search heuristics. Sci Rep 8(1):13694

Karafotias G, Hoogendoorn M, Eiben ÁE (2015) Parameter control in evolutionary algorithms: trends and challenges. IEEE Trans Evol Comput 19(2):167–187

Kaufmann KW, Lemmon GH, DeLuca SL, Sheehan JH, Meiler J (2010) Practically useful: what the Rosetta protein modeling suite can do for you. Biochemistry 49(14):2987–2998

Kihara D (2014) Protein structure prediction. Springer, Berlin

Kim DE, Blum B, Bradley P, Baker D (2009) Sampling bottlenecks in de novo protein structure prediction. J Mol Biol 393(1):249–260

Lee J, Freddolino PL, Zhang Y (2017) Ab initio protein structure prediction. In: From protein structure to function with bioinformatics. Springer, Berlin, Germany, pp 3–35

Li B, Chiong R, Lin M (2015) A balance-evolution artificial bee colony algorithm for protein structure optimization based on a three-dimensional AB off-lattice model. Comput Biol Chem 54:1–12

Li Z, Scheraga HA (1987) Monte Carlo-minimization approach to the multiple-minima problem in protein folding. Proc Natl Acad Sci 84(19):6611–6615

Lopes HS (2008) Evolutionary algorithms for the protein folding problem: a review and current trends. In: Computational intelligence in biomedicine and bioinformatics. Springer, pp 297–315

McGuffin LJ, Bryson K, Jones DT (2000) The PSIPRED protein structure prediction server. Bioinformatics 16(4):404–405

Narloch PH, Dorn M (2019) A knowledge based self-adaptive differential evolution algorithm for protein structure prediction. In: International conference on computational science. Springer, pp 87–100

Narloch PH, Parpinelli RS (2016) Diversification strategies in differential evolution algorithm to solve the protein structure prediction problem. In: International conference on intelligent systems design and applications. Springer, Porto, Portugal, pp 125–134

Narloch PH, Parpinelli RS (2017) The protein structure prediction problem approached by a cascade differential evolution algorithm using rosetta. In: 2017 Brazilian conference on intelligent systems (BRACIS). IEEE, Uberlandia, Brazil, pp 294–299

Nunes LF, Galvão LC, Lopes HS, Moscato P, Berretta R (2016) An integer programming model for protein structure prediction using the 3D-HP side chain model. Discret Appl Math 198:206–214

Oliveira M, Borguesan B, Dorn M (2017) Sade-spl: a self-adapting differential evolution algorithm with a loop structure pattern library for the psp problem. In: Evolutionary computation (CEC). 2017 IEEE Congress on. IEEE, Donostia - San Sebastián, Spain, pp 1095–1102

Olson B, Shehu A (2012) Efficient basin hopping in the protein energy surface. Bioinformatics and Biomedicine (BIBM). In: 2012 IEEE International conference on. IEEE, USA, pp 1–6

Parpinelli RS, Plichoski GF, Da Silva RS, Narloch PH (2019) A review of technique for on-line control of parameters in swarm intelligence and evolutionary computation algorithms. Int J Bio-Inspir Comput 13:1–20

Prentiss MC, Wales DJ, Wolynes PG (2008) Protein structure prediction using basin-hopping. J Chem Phys 128(22):06B608

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: Evolutionary Computation, 2005, vol 2. The 2005 IEEE Congress on. IEEE, Edinburgh, Scotland, pp 1785–1791

Ramyachitra D, Ajeeth A (2017) Modcsa-ca: a multi objective diversity controlled self adaptive cuckoo algorithm for protein structure prediction. Gene Rep 8:100–106

Rohl CA, Strauss CE, Misura KM, Baker D (2004) Protein structure prediction using rosetta. In: Methods in enzymology, vol. 383. Elsevier, New York, USA, pp 66–93

Salomon-Ferrer R, Case DA, Walker RC (2013) An overview of the amber biomolecular simulation package. Wiley Interdiscip Rev Comput Mol Sci 3(2):198–210

Silva RS, Parpinelli RS (2018) A multistage simulated annealing for protein structure prediction using Rosetta. Anais do Computer on the Beach pp 850–859

Silva RS, Parpinelli RS (2019) A self-adaptive differential evolution with fragment insertion for the protein structure prediction problem. In: International workshop on hybrid metaheuristics. Springer, pp 136–149

Simoncini D, Schiex T, Zhang KY (2017) Balancing exploration and exploitation in population-based sampling improves fragment-based de novo protein structure prediction. Proteins Struct Funct Bioinf 85(5):852–858

Sinha A, Malo P, Deb K (2018) A review on bilevel optimization: from classical to evolutionary approaches and applications. IEEE Trans Evol Comput 22(2):276–295

Song S, Ji J, Chen X, Gao S, Tang Z, Todo Y (2018) Adoption of an improved PSO to explore a compound multi-objective energy function in protein structure prediction. Appl Soft Comput 72:539–551

Sudha S, Baskar S, Amali SMJ, Krishnaswamy S (2015) Protein structure prediction using diversity controlled self-adaptive differential evolution with local search. Soft Comput 19(6):1635–1646

Tawhid MA, Ali AF (2017) A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. Memetic Comput 9:347–359. https://doi.org/10.1007/s12293-017-0234-5

Varela D, Santos J (2019) Crowding differential evolution for protein structure prediction. In: International work-conference on the interplay between natural and artificial computation. Springer, pp. 193–203

Vlachakis D, Bencurova E, Papangelopoulos N, Kossida S (2014) Current state-of-the-art molecular dynamics methods and applications. In: Advances in protein chemistry and structural biology, vol. 94. Elsevier, New York, USA, pp 269–313

Walsh G (2002) Proteins: biochemistry and biotechnology. John Wiley & Sons

Wilk MB, Shapiro S (1968) The joint assessment of normality of several independent samples. Technometrics 10(4):825–839

Zaman AB, Shehu A (2019) Balancing multiple objectives in conformation sampling to control decoy diversity in template-free protein structure prediction. BMC Bioinf 20(1):211

Zemla A (2003) LGA: a method for finding 3D similarities in protein structures. Nucleic Acids Res 31(13):3370–3374