SANTA CATARINA STATE UNIVERSITY – UDESC
CENTER OF TECHNOLOGICAL SCIENCES – CCT
GRADUATE PROGRAM IN APPLIED COMPUTING – PPGCA

FELIPE MARCHI RAMOS

# A MULTI-OBJECTIVE BIASED RANDOM-KEY GENETIC ALGORITHM APPLIED TO THE PROTEIN STRUCTURE PREDICTION PROBLEM

JOINVILLE
2021

**FELIPE MARCHI RAMOS**

# A MULTI-OBJECTIVE BIASED RANDOM-KEY GENETIC ALGORITHM APPLIED TO THE PROTEIN STRUCTURE PREDICTION PROBLEM

Master thesis submitted to the Computer Science Department at the College of Technological Science of Santa Catarina State University in fulfillment of the partial requirement for the Master's degree in Applied Computing.

Advisor: Rafael Stubs Parpinelli

**JOINVILLE**
**2021**

Ramos, Felipe Marchi
A Multi-objective Biased Random-Key Genetic Algorithm Applied to the Protein Structure Prediction Problem / Felipe Marchi  Ramos. - Joinville, 2021.
87 p. : il. ; 30 cm.

Advisor:  Rafael Stubs  Parpinelli.
Thesis (Master) - Santa Catarina State University, Center of Technological Sciences, Graduate Program in Applied Computing, Joinville, 2021.

1. Protein structure prediction. 2. Multi-objective optimization. 3. Evolutionary algorithms. 4. Secondary structures. 5. Contact maps. I.  Parpinelli, Rafael Stubs . II. , . III. Santa Catarina State University, Center of Technological Sciences, Graduate Program in Applied Computing. IV. Title.

**FELIPE MARCHI RAMOS**

# A MULTI-OBJECTIVE BIASED RANDOM-KEY GENETIC ALGORITHM APPLIED TO THE PROTEIN STRUCTURE PREDICTION PROBLEM

Master thesis submitted to the Computer Science Department at the College of Technological Science of Santa Catarina State University in fulfillment of the partial requirement for the Master's degree in Applied Computing.

Advisor: Rafael Stubs Parpinelli

**EXAMINATION BOARD:**

Dr. Rafael Stubs Parpinelli
UDESC-CCT (president/advisor)

Dr. Yuri Kaszubowski Lopes
UDESC-CCT

Dr. Marcio Dorn
UFRGS

Joinville, 25th, November of 2021

# ABSTRACT

Proteins are base molecules present in live organisms. The functions of a protein are related to its structure. Therefore understanding the structure of a protein is necessary to understand its function. Protein structures are formed by complex biological processes that are still not entirely understood. The protein structure prediction problem is one of the most important bioinformatics problems. Computational methods can be used to solve this problem. Among the different types of methods are the *de novo* methods, which are able to generate protein structures without the need of having known similar structures to the predicted protein. These methods transform the prediction problem into an optimization problem, using optimization models that combine different energy functions and high-level information. These models usually have only a single optimization objective. However, it is known that this single objective optimization approach may harm the optimization search due to the existence of conflicts between the different terms that compose the optimization objective. In this regard, this work aims to propose a multi-objective model and optimization method for the proteins structure prediction problem. The proposed model has three objectives: energy function, secondary structure, and contact maps. The selected optimization model was the Biased Random-Key Genetic Algorithm (BRKGA), which was modified to optimize multi-objective problems (MO-BRKGA) and employs online parameter control. The final predictor comprises two phases of the MO-BRKGA and selects a final structure using the MUFOLD-CL clustering method. Three experiments were performed to evaluate the predictor performance. These experiments demonstrated the power of the proposed predictor, which generated highly competitive results with the literature.

**Keywords**: Protein structure prediction. Multi-objective optimization. Evolutionary algorithms. Secondary structures. Contact maps.

# RESUMO

Proteínas são moléculas base em organismos vivos. O funcionamento de uma proteína está associado a sua estrutura, portanto entender ela é fundamental para entender seu funcionamento. Estruturas de proteínas são formadas por processos biológicos complexos, que até então não são totalmente conhecidos. O problema da predição de estruturas de proteínas é um dos problemas principais da bioinformática. Métodos computacionais podem ser usados para resolver esse problema. Entre os diferentes tipos de métodos computacionais, estão os métodos *de novo*, que não exigem a existência de estruturas conhecidas similares à proteina predita. Esses métodos transformam a predição em um problema de otimização, utilizando modelos que combinam funções de energia com informações de alto nível. Esses modelos geralmente apresentam apenas um único objetivo, porém se sabe que o uso de um único objetivo pode prejudicar a otimização devido a existência de conflitos entre diferentes termos da função a ser otimizada. Nesse sentido, esse trabalho tem por objetivo propor um modelo multiobjetivo e método de otimização para o problema de predição de estruturas de proteínas. O modelo é composto por 3 objetivos: função energia, estruturas secundárias, e mapas de contato. O método de otimização utilizado foi o *Biased Random-Key Genetic Algorithm* (BRKGA), modificado para conseguir otimizar problemas multi-objetivo (MO-BRKGA) e aplica controle online de parâmetros. O preditor final é composto de um otimizador, que utiliza duas fases do MO-BRKGA e escolhe uma solução final utilizando o método de clustering MUFOLD-CL. Três tipos de experimentos foram realizados para analisar o desempenho do método e do modelo proposto. Os experimentos demonstraram o potencial do método proposto, que foi capaz de encontrar soluções altamente competitivas com a literatura.

**Palavras-chave**: Predição de estruturas de proteínas. Otimização multi-objetivo. Algoritmos evolucionários. Estruturas secundárias. Mapa de contato.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| BRKGA | Biased Random-Key Genetic Algorithm |
| CASP | Critical Assessement of Structure Prediction |
| CHARMM | Chemistry at HARvard Macromolecular Mechanics |
| GA | Genetic Algorithms |
| GDT-TS | Global Distance Test Total Score |
| MO-BRKGA | Multi-Objective Biased Random-Key Genetic Algorithm |
| MOEA | Multi-objective Evolutionary Algorithm |
| NMR | Nuclear Magnetic Resonance |
| PDB | Protein Databank |
| PSP | Protein Structure Prediction |
| RMSD | Root-Mean-Square Deviation |

## LIST OF SYMBOLS

| | |
|---|---|
| $\phi$ | Phi angle |
| $\psi$ | Psi angle |
| $\omega$ | Omega angle |
| Å | Angstrom |
| $C_\alpha$ | Carbon alpha |
| $L$ | Amino acid sequence length |
| $d$ | Euclidean distance |
| $F$ | Pareto frontier |
| $P$ | Population |
| $N_{it}$ | Number of iterations |
| $p$ | Population size |
| $c_{pr}$ | Crossover probability |
| $p_e$ | Elite fraction |
| $p_m$ | Random fraction |
| $p_d$ | Diversified elite fraction |
| $\delta$ | Exploration diversity threshold |

CONTENTS

# 1 INTRODUCTION

Proteins are base molecules present in living organisms (GARRET; GRISHAM, 2010). They are responsible for many biological functions, and understanding their mechanisms is essential to have a better understanding of living beings. One application of this knowledge about proteins is in the biomedicine and pharmaceutics areas (DILL et al., 2008), where novel proteins could be developed to combat particular types of diseases.

One way to understand how proteins work is to study their spatial structure, which determines many of their functions (GU; BOURNE, 2009). One of the main factors that determine the spatial structure of a protein is the amino acid sequence. However, the amino acid sequence is not the only factor defining the spatial structure. The process in which an amino acid chain is transformed into the final spatial structure is called folding, and it is regarded as a very complex process, composed of environmental factors and chemical and physical interactions (DILL et al., 2008). The prediction of protein structures is called the Protein Structure Prediction (PSP) problem and it is considered one of the most important objectives of computational biology (DILL et al., 2008).

## 1.1 PROBLEM

The PSP problem has the objective of determining the spatial structure of proteins. Although these structures can be determined using classical laboratory methods, such as Nuclear Magnetic Resonance (NMR) or X-ray crystallography (GU; BOURNE, 2009), they have a considerable cost and are time-consuming. An alternative method is the use of computational resources to simulate these structures.

In the literature, there are many computational methods proposed as possible approaches to the PSP problem, such as those reviewed by Dorn et al. (2014) and Márquez-Chamorro et al. (2015). Among these methods, there are also works that further explore the use of specific types of protein structure information (SILVA; PARPINELLI, 2019; WILL; PARPINELLI, 2020). However, the PSP is still considered to be an open problem. So far, there is no viable general solution to this challenging problem.

Among the different types of computational methods applied to this problem in the literature, there are the *ab initio* methods. Different from other types of methods, the *ab initio* methods only need as input the amino acid sequence to work (GU; BOURNE, 2009). These methods work by generating structures using some evaluation function, which guides the optimization towards the optimal structure.

Using these evaluation functions, the *ab inito* methods reduce the PSP problem to a mathematical model optimization problem. To generate adequate structures, these models must use enough information to simulate the natural process of protein structure formation, called folding (DILL et al., 2008). This process, however, is considerably complex and is hard to simulate using accurate methods because of its high computational complexity.

Energy functions are used to approximate the folding process. These functions try to simulate the physical and chemical interactions that occur in the natural environment. Standard energy functions are used in combination of force fields (GU; BOURNE, 2009), which are parameters derived from experimental data and are used to model the atomic interactions. As they use experimental data, these energy functions may not consider all necessary information to generate accurate structures.

More information can be incorporated into the evaluation function to increase its efficacy. This extra information may complement the used energy function by adding more atomic interactions, but it can also use high-level knowledge about the protein structure. Combining different types of information in the evaluation fitness allows a more guided optimization and accurate generation of structures.

For example, a high-level knowledge about a protein structure is the information about its secondary structure. This information can be predicted from existing proteins and can be incorporated into the optimization model to complement the energy function. The combination of the energy function with this external information is called a *de novo* method, which is a class of knowledge-based methods that generalizes the *ab initio* methods (which only consider the energy function) (GU; BOURNE, 2009).

As *de novo* methods optimize a mathematical model, it can be said that the evaluation function represents an optimization objective. Usually, the optimization models proposed for the PSP problem use a single objective as the evaluation function. However, it is possible to model the PSP as a multi-objective model. The main objective of this work is to propose an *de novo* multi-objective approach to the PSP problem.

## 1.2 MOTIVATION

Considering the single-objective optimization model for the PSP problem, the energy function is commonly used as an optimization objective. These energy functions are composed of multiple terms that represent different information about the physical and chemical interactions.

Although it is possible to combine multiple information into a single objective, it is not always optimal. It is known that some of the terms that compose an energy function, such as the bonded and non-bonded energies, are in conflict (CUTELLO; NARZISI; NICOSIA, 2006). These conflicts indicate that optimizing one particular term of the function may not optimize the others. To better optimize the model, it is interesting to separate conflicting terms in different objectives.

This separation of conflicting terms creates a multi-objective model, whose final solution is a set of non-dominating solutions (KALYANMOY; DEB, 2001). This set, also called the *Pareto set*, represents possible solutions for the mathematical model. As each objective is optimized independently, a multi-objective model minimizes the impact of possible conflicts.

In the context of PSP, a multi-objective model may divide an energy function into separate objectives, and it can also add new objectives representing different types of information about the protein structure, such as the information about secondary structures.

An acceptable optimization method must be used to optimize a multi-objective model. One class of optimization methods suitable for multi-objective problems is the evolutionary algorithms (KALYANMOY; DEB, 2001). These algorithms use the concept of populations of solutions, in which each iteration of the algorithm optimizes a set of solutions. In the end, the optimized solution can be taken either from the final population or from some external solution set, usually denominated *archive*.

In multi-objective problems, the optimal solution is a set of solutions. As evolutionary algorithms use populations (set of solutions), it is natural to optimize a multi-objective problem. One member of this class of algorithms is the Biased Random-Key Genetic Algorithm (BRKGA) (GONÇALVES; RESENDE, 2011).

The BRKGA is an algorithm that differs from others because of its standardized structure. The method has a clear separation between the problem-dependent and problem-independent parts. As a result, problem-independent components, such as genetic operators, are fixed and do not need to be modified. This separation allows the developer to focus more on the problem modeling and less on modeling the optimization algorithm itself.

## 1.3 OBJECTIVES

The primary objective of this work is the development of a multi-objective model and optimizer for the PSP problem. As stated in Section 1.2, the PSP problem displays a structure that may be better explored as a multi-objective problem. Given this, a multi-objective model is proposed for the problem.

This work proposes the use of the BRKGA method as an optimizer of this model. This algorithm was not found in the literature for the multi-objective PSP. Given the reasons stated in Section 1.2, the BRKGA is an interesting algorithm that can be applied to the PSP problem.

As the optimized problem has a multi-objective model, the final solution is a set. However, the solution for the problem is a single solution, which represents the protein structure. To be able to select a single solution from this optimized set, a decision-making step is necessary. In this work, the MUFOLD-CL (ZHANG; XU, 2013) clustering method was used to select a final solution from the optimized set of solutions.

The specific objectives of this work are:

- Develop an optimization model using potential energy, secondary structure, and contact map information for the protein structure prediction problem;
- Develop a multi-objective BRKGA with parameter control to optimize the proposed model;
- Compare experiments results with the state-of-the-art of *de novo* methods.

## 1.4 LIST OF PUBLICATIONS

- MARCHI, Felipe; PARPINELLI, Rafael Stubs. A multi-objective approach to the proteinstructure prediction problem using the biased random-key genetic algorithm. In: IEEE. **2021 IEEE Congress on Evolutionary Computation (CEC)**. New York, 2021. p. 1070–1077.

## 1.5 DOCUMENT STRUCTURE

This document is organized as follows: Chapter 2 presents theoretical concepts used through the work. Chapter 3 presents related works. Chapter 4 presents the proposed methodology, and Chapter 5 presents the experiments and results of this work. Finally, Chapter 6 concludes the work, presenting final remarks and pointing possible future works.

# 2 THEORETICAL FOUNDATIONS

## 2.1 PROTEINS

Proteins are biomolecules composed of amino acids linked by peptide bonds. These peptide bonds are the connection between the amino and carboxyl group (GARRET; GRISHAM, 2010). Proteins are responsible for many biological processes. The 20 most common amino acids found in proteins are shown in Table 1. The table displays the full name of each amino acid and also the three and one-letter codes, which can be useful in some contexts (GARRET; GRISHAM, 2010).

Table 1 – Natural amino acids

| Amino Acid | Three-letter Code | One-letter Code |
|---|---|---|
| Alanine | Ala | A |
| Arginine | Arg | R |
| Asparagine | Asn | N |
| Aspartic acid | Asp | D |
| Cysteine | Cys | C |
| Glutamic acid | Glu | E |
| Glutamine | Gln | Q |
| Glycine | Gly | G |
| Histidine | His | H |
| Isoleucine | Ile | I |
| Leucine | Leu | L |
| Lysine | Lys | K |
| Methionine | Met | M |
| Phenylalanine | Phe | F |
| Proline | Pro | P |
| Serine | Ser | S |
| Threonine | Thr | T |
| Trypyophan | Trp | W |
| Tyrosine | Tyr | Y |
| Valine | Val | V |

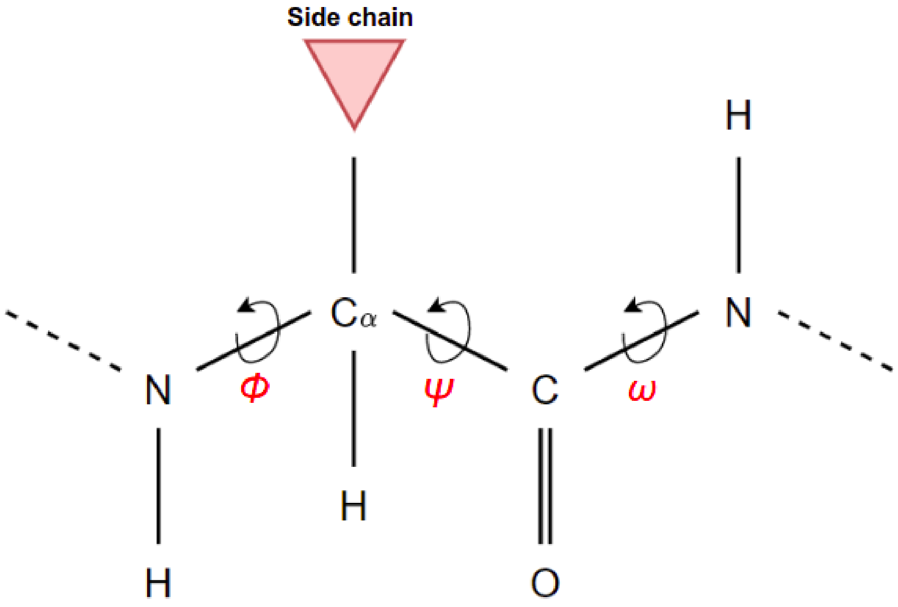
Source: Author

The amino acids are molecules composed of a central carbon, called $\alpha$-carbon, that is connected to an amino group, a carboxyl group, a hydrogen molecule, and a side chain (GARRET; GRISHAM, 2010). The side chain is a variable group that is unique to each type of amino acid. The molecular structure of each amino acid is the same, changing only the side chain, which gives its identity. The structure of a generic amino acid can be seen in Figure 1. Also present in the figure are the torsion angles $\phi$, $\psi$, and $\omega$, which will be useful later to represent a protein computationally.

A sequence of linked amino acids results in a protein, which can belong to one of three global classes (GARRET; GRISHAM, 2010):

- **Globular proteins**: exhibit compact structures where the polypeptide chain folds to hide the hydrophobic amino acids and expose the hydrophilic;
- **Fibrous proteins**: exhibit regular and linear structures; and
- **Membrane proteins**: exhibit distinct structure to associate with various membrane systems in cells.

Figure 1 – Chemical structure of an amino acid. All natural amino acids exhibit this chemical structure. The side chain is a chemical structure that is unique for each amino acid type that is displayed in Table 1



Source: Author

Also, the protein structure can be described using a hierarchical organization with four levels: primary, secondary, tertiary, and quaternary.

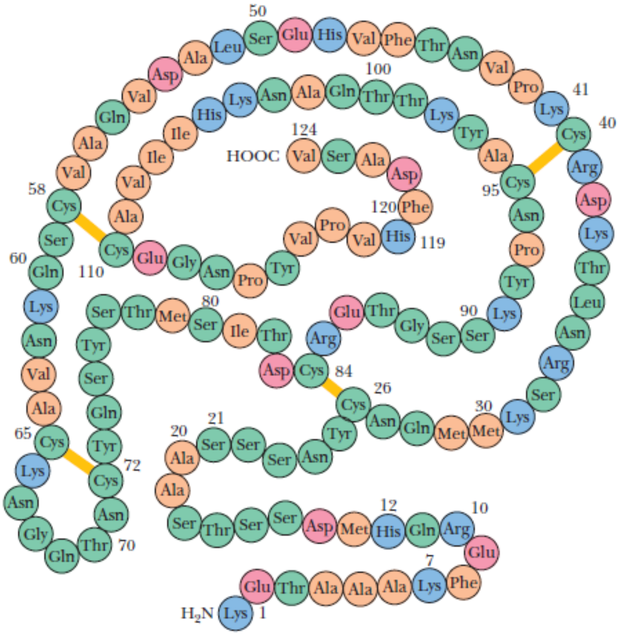
### 2.1.1 Primary structure

The primary structure of a protein is the amino acid sequence itself (GARRET; GRISHAM, 2010). An example of primary structure can be seen in Figure 2.

### 2.1.2 Secondary structure

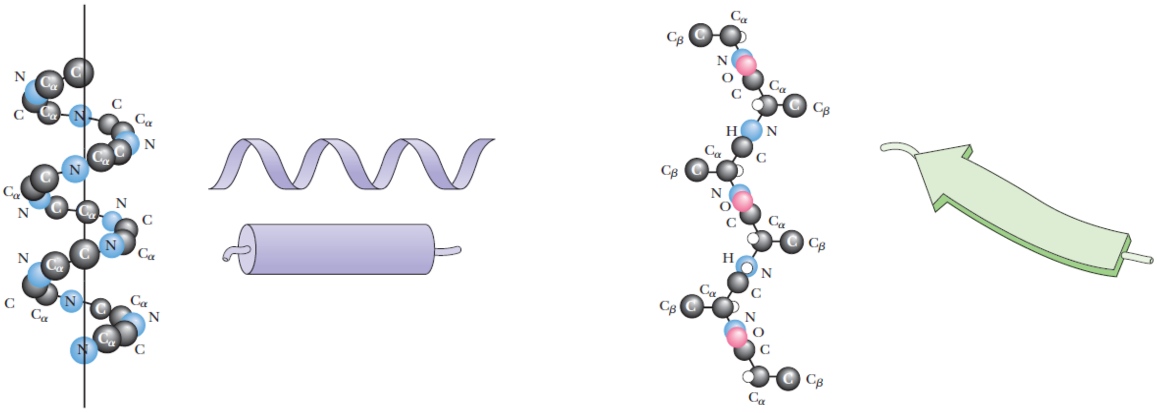
Secondary structures are regular structural patterns that appear on many proteins structures (GARRET; GRISHAM, 2010). These regular structures are usually classified as either $\alpha$**-helices** or $\beta$**-sheets**, interconnected by irregular structures called **loops** or **coils**. Example of these structures can be seen on Figure 3.

Figure 2 – Primary structure of a protein. Each amino acid is represented by a circle with the 3-letter identifier. The numbers represent the sequence position. The lines connecting some pairs of amino acids are disulfide bridges, present in some proteins.



Source: Garret e Grisham (2010)

Figure 3 – Secondary structures $\alpha$-helix (left) and $\beta$-sheet (right). It is shown for both structures the atomic structure and the ribbon representation. The ribbon representation of protein structures is popular as it simplifies and distinguishes the secondary structures.

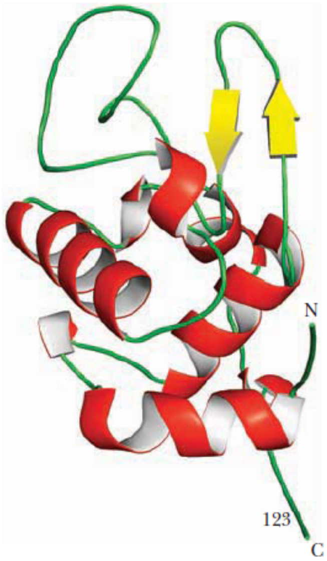


Source: Garret e Grisham (2010)

### 2.1.3 Tertiary structure

The complete three-dimensional structure of a protein is called its tertiary structure (GARRET; GRISHAM, 2010). The process that creates this structure is called protein folding. An example of tertiary structure can be seen in Figure 4.

Figure 4 – Tertiary structure of a protein (ribbon representation). It is possible to identify the three main secondary structure types: the coils (green), $\alpha$-helices (red), and the $\beta$-sheets (yellow).

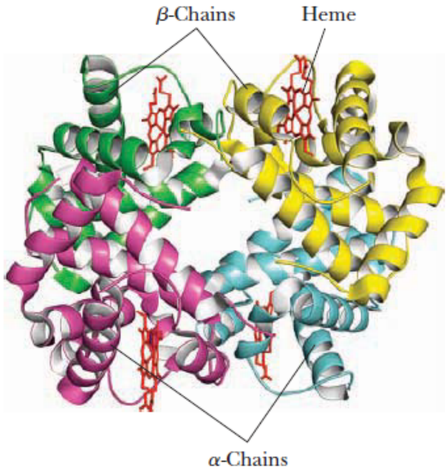


Source: Garret e Grisham (2010)

### 2.1.4 Quaternary structure

Proteins can be composed of multiple independent polypeptide chains, which interact between themselves and form the called quaternary structure (GARRET; GRISHAM, 2010). Each polypeptide chain, referred to as a subunit, has its tertiary structure. An example of this type of structure is shown in Figure 5.

Figure 5 – Quaternary structure of protein hemoglobin, which has 4 subunits. Each subunit is an individual protein structure, and, in this case, they are all formed only by $\alpha$-helices.



Source: Garret e Grisham (2010)

### 2.1.5 Computational representation

There are several ways to represent a protein structure computationally. The most direct would be to maintain complete information on all the atoms and interactions of the structure. However, this type of representation can be computationally expensive given the large number of atoms and interactions of a single protein structure (GU; BOURNE, 2009).

Simplifications can be made to this full atom representation, generating a trade-off between accuracy and computational complexity. Some of the more common simplifications involve the solvent model, which is usually implicit, and the side chains (GARRET; GRISHAM, 2010). There are also simplified models that limit the protein structure to a lattice, which considerably reduce the conformation space of search algorithms but also reduce the structure accuracy.

One interesting computational representation of protein structures uses only the backbone and side chain torsion angles of each amino acid. By using these angles, the final three-dimensional structure can be uniquely determined. Instead of maintaining the entire atomic structure of each amino acid, algorithms may use only the $\phi$, $\psi$, and $\omega$ backbone angles and the $\chi$ side chain angles. These angles can be seen in Figure 1.

## 2.2 PROTEIN STRUCTURE PREDICTION

One way to study proteins is through the study of their spatial structure. The initial methods of generating protein structure were done through laboratory experiments. These methods can generate protein structures with high precision for some proteins. However, they are slow and expensive, and as the amount of unknown protein structures is considerably large, more efficient structure generation methods are necessary.

One possibility is the use of computational resources to generate these structures. Even though the folding process is still largely unknown, several types of algorithms can predict protein structures. Among these algorithms, three main paradigms can be defined: homology, threading, and *ab initio* (or *de novo*) methods (GU; BOURNE, 2009).

Homology methods can predict protein structures using known protein structures that are homologous to the protein to be predicted (GU; BOURNE, 2009). These methods can generate structures with considerable accuracy. However, it can only be applied if there exist known homologous structures.

Threading methods also need information about known structures, although they do not need to be direct homologous. These methods instead work with parts of the amino acid sequence, finding distant homologies (GU; BOURNE, 2009). Although it is possible to work with proteins without direct known homologous, there is still a reliance on data about known protein structures.

*Ab initio* methods can generate structures without using data about known structures. These methods use physical energy functions to guide the generation of protein structures (GU;

BOURNE, 2009), using only the amino acid as necessary input. As such, *ab initio* methods effectively transform the PSP problem into an optimization problem.

This optimization problem usually consists of finding the protein structure with minimal energy. To this end, search methods were developed to generate protein structures using some energy function. As such, *ab initio* methods consist in defining a suitable representation for the protein structure and a suitable energy function to guide the optimization process.

As presented in Section 2.1.5, there are several ways of representing a protein structure computationally. The choice of representation has to consider the trade-off between accuracy and computational complexity.

Besides the computational representation of a protein structure, defining a function to evaluate these structures is also necessary. As *ab initio* methods generally perform searches in the space of structures, it is necessary to select a suitable evaluation function. Regarding accuracy, the most optimal choice would be an evaluation function that represents the natural energy function. However, such realistic energy functions are computationally expensive (DILL et al., 2008).

Similar to selecting a structure representation, there is a trade-off between accuracy and cost when selecting an evaluation function. Simplified molecular mechanics energy functions are usually used as evaluating functions. These energy functions consider a set of intramolecular interactions parameterized by experimental data (force fields) to approximate the energy of a particular protein structure (GU; BOURNE, 2009).

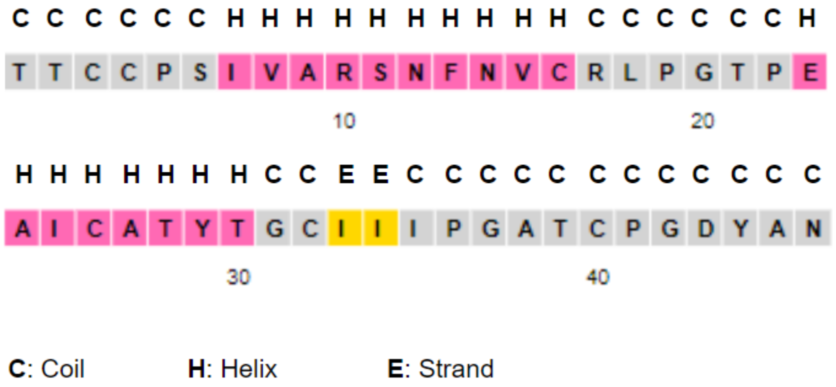
Other types of information can be incorporated into the evaluation function to enhance the search process. Although pure *ab initio* methods do not use information from known structures, it may be beneficial to use this type of information, if available. There are different types of high-level information that can be added to the evaluation function, such as *secondary structure prediction* and *contact maps*. When an *ab initio* method incorporates information other than the energy function, it is called an *de novo* method (GU; BOURNE, 2009).

### 2.2.1 Secondary structure prediction

The complete structure of proteins with a single polypeptide chain is the tertiary structure. This structure can be seen as the union of different secondary structures, the most common being $\alpha$-helices, $\beta$-sheets, and coils (GARRET; GRISHAM, 2010). By having the information of which secondary structures compose the protein structure, it is possible to refine the search process.

Secondary structure prediction methods, such as PSIPRED (JONES, 1999), use the amino acid sequence as input and output information about possible secondary structures present in the tertiary structure. It is possible to enhance the evaluation function, such as penalizing structures with secondary structures different from the predicted. An example of secondary structure prediction can be seen in Figure 6, where for each amino acid, it is predicted to which secondary structures it belongs.

Figure 6 – Example of secondary structure prediction for the 1CRN protein. The amino acids are represented by the 1-letter code. For each amino acid it is predicted a secondary structure, which can be coil (C), helix (H), or strand (E). A color is also associated to each type of secondary structure. The numbers represent the position numbers, in increments of 10.


C C C C C C H H H H H H H H H H C C C C C C H
T T C C P S I V A R S N F N V C R L P G T P E
10
20
H H H H H H H C C E E C C C C C C C C C C C C
A I C A T Y T G C I I I P G A T C P G D Y A N
30
40
C: Coil
H: Helix
E: Strand

Source: Author

In Figure 6, an amino-acid sequence is described using three types of secondary structures: coils (C), $\alpha$-helices (H), and $\beta$-sheets (E). The amino acids are described by their one-letter code, which can be seen in Table 1. This example is a prediction of secondary structure for the protein 1CRN, using the PSIPRED server [1].

### 2.2.2 Contact maps

Another type of high-level information about protein structures is the contact maps. These maps represent contacts between pairs of amino acids in the folded structure. Two pairs are in contact if their spatial distance is less than some predefined threshold, which is usually 8Å[2] (LENA; NAGATA; BALDI, 2012).

Similar to the prediction of secondary structures, it is possible to use known structures to predict contact maps of some unknown protein, using contact map predictors such as the CoinDCA method (MA et al., 2015). With these contacts, it is possible to enhance the evaluation function and refine the search space. One possible way would be to penalize pairs of amino acids in the generated structure that are not in contact but were predicted to be and vice-versa.

As the prediction of a contact map is not exact, it is common to use just a subset of the best contacts. These subsets are usually defined by a fraction of the amino acid sequence length (MONASTYRSKYY et al., 2014): $L/2$, $L/5$, $L/10$, etc., where $L$ is the amino acid sequence length. Contacts are selected by their quality, measured by their confidence score, that is, the probability of the contact being correctly predicted.

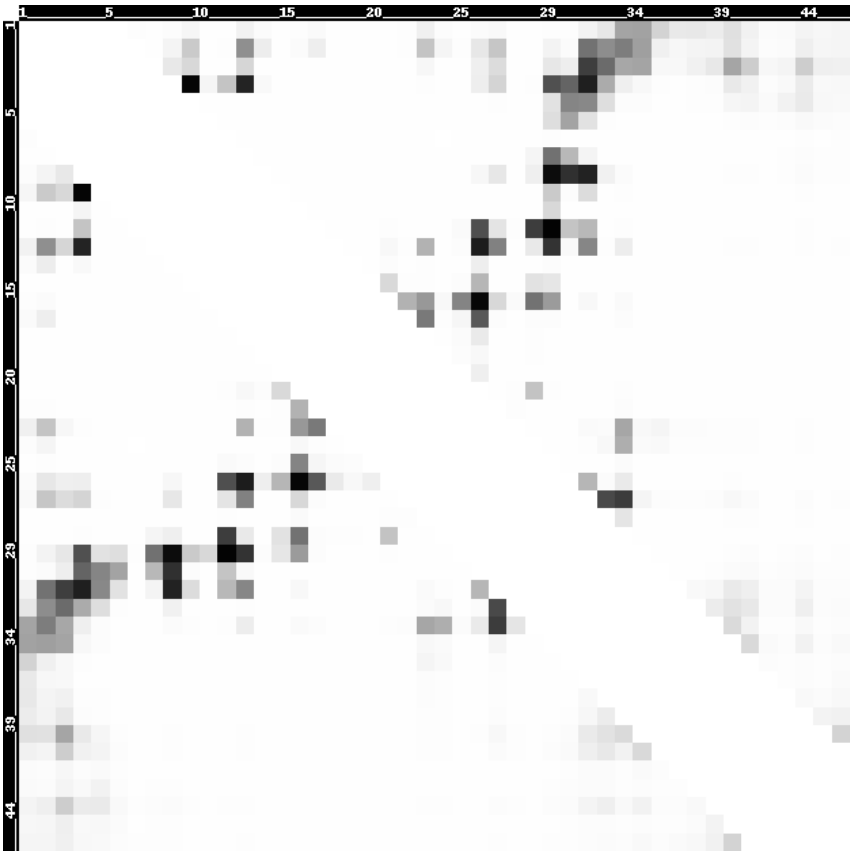
An example of contact map prediction is shown in Figure 7. The contact map is a matrix $L \times L$, where $L$ is the amino acid sequence length. For each pair, a confidence score is given

[1] http://bioinf.cs.ucl.ac.uk/psipred/

[2] Angstrom (Å) is a unit of length, equal to $10^{-10}$ meter.

between 0 and 1. In this figure, scores closer to 1 are darker, indicating a high confidence score for these pairs of amino acids. This contact map is a prediction for the 1CRN protein, using the RaptorX server[3] (CoinDCA (MA et al., 2015)).

Figure 7 – Example of contact map prediction for the 1CRN protein. The contact map is a *LxL* matrix where each position represents a pair of amino acids. Each pair receives a value between 0 and 1, which is the confidence score for that pair. The greater the value, the higher the probability of that pair being true in the native structure. In this figure, higher values have darker colors.



Source: Author

### 2.2.3 Quality of predicted structures

The Root Mean Square Deviation (RMSD) and Global Distance Test total score (GDT_TS) metrics can be utilized to measure the quality of the predicted structures. The RMSD measures the distance of atoms between two superimposed structures, with lower values indicating higher similarity structures (MAIOROV; CRIPPEN, 1994). The distance is taken considering the $C_\alpha$ atoms (protein backbone) and can be mathematically defined as:

$$RMSD(\vec{a},\vec{b}) = \sqrt{\frac{\sum_{i=1}^{L} d(a_i,b_i)^2}{L}} \tag{1}$$

[3] http://raptorx.uchicago.edu/

where $L$ is the number of residues (amino acids), $\vec{a}$ and $\vec{b}$ are two superimposed structures, and $d(a_i, b_i)$ returns the Euclidean distance (in angstroms) between the atoms $a_i$ and $b_i$.

The GDT_TS is another similarity metric and is considered more stable than the RMSD regarding local regions (ZEMLA, 2003). It can mathematically be defined as:

$$GDT\_TS(\vec{a},\vec{b}) = \frac{100 \times (S_1 + S_2 + S_3 + S_4)}{4L} \tag{2}$$

where $L$ is the number of residues, and $\vec{a}$ and $\vec{b}$ are superimposed structures. $S_1$, $S_2$, $S_3$, and $S_4$ are the number of residues aligned in the superimposition with distance under 1Å, 2Å, 4Å, and 8Å, respectively. Higher values of the metric indicate higher similarity between structures. As with the RMSD, only the $C_\alpha$ atoms are usually considered.

## 2.3 ROSETTA FRAMEWORK

The Rosetta Software Suite[4] is a framework with software libraries for macromolecule modeling. The framework is composed of protocols that can be used to accomplish the analysis, design, and prediction of complex bio-molecular systems. One of these protocols is the Rosetta *de novo* algorithm, which is a Monte Carlo method (ROHL et al., 2004).

The Rosetta framework also has energy functions for both the all atoms and centroid models. The all atoms energy function is a single force field composed of several terms representing interactions (ROHL et al., 2004), similar to others force fields such as CHARMM or GROMOS. The centroid energy function, however, is divided into multiple energy functions (score0, 1, 2, 3, ...), each composed of some interaction terms and used for different purposes (ROHL et al., 2004).

Another important feature of the Rosetta framework is the fragment library. Fragments are local structures extracted from known protein structures similar to the one being predicted (ROHL et al., 2004). The framework has a protocol for fragment generation, which generates fragments of different sizes, the most commonly utilized being sizes 3 and 9. The Rosetta *de novo* protocol uses fragment insertion as search mechanism, creating initial models by combining different fragments (ROHL et al., 2004).

## 2.4 MULTI-OBJECTIVE OPTIMIZATION

In a multi-objective model, multiple objectives must be optimized. In these models, conflicts between objectives may occur, where the optimization of one objective will result in the deterioration of the others. Unlike single-objective models, the optimal solution of a multi-objective model is not a single solution unless there is no conflict between any objective. If there are conflicts, the optimal solution will be a set of non-dominating solutions (KALYANMOY; DEB, 2001).

[4] https://www.rosettacommons.org/support/overview

A multi-objective model can be mathematically described as:

$$\min \quad f_i(\vec{x}), \quad i \in M_1 \tag{3}$$

$$\max \quad f_i(\vec{x}), \quad i \in M_2 \tag{4}$$

$$\text{subject to } \vec{x} \in X \tag{5}$$

where $\vec{x}$ represents the solution vector composed of $n$ problem variables, notations (3) and (4) represent objective functions that have to be minimized or maximized, respectively, with $M_1$ and $M_2$ being sets of functions of each type. The set $X$ in notation (5) is the set of feasible solutions, which may also be defined by some constraint functions.

Considering that some objectives of the model conflict, optimizing such model will result in a set of **non-dominated solutions**. The dominance criteria $\vec{x} \preceq \vec{y}$, where solution $\vec{x}$ dominates $\vec{y}$, can be defined as (KALYANMOY; DEB, 2001):

1. For each objective, solution $\vec{x}$ is no worse than $\vec{y}$; and
2. There exists at least one objective where solution $\vec{x}$ is better than $\vec{y}$.

With the definition of the dominance of solutions, it is possible to define the concept of **Pareto-optimality**. A Pareto set is any set of non-dominated solutions, with the Pareto-optimal set being the set of non-dominated solutions considering the entire search space (KALYANMOY; DEB, 2001). As such, the Pareto-optimal set can be considered as the optimal solution to a multi-objective problem.

## 2.5 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Evolutionary algorithms use biological evolution as a metaphor to describe generic optimization frameworks (EIBEN; SMITH et al., 2003), also known as meta-heuristics. These frameworks utilize the concept of population, where each individual of the population is a candidate solution. These individuals are evolved through selection mechanisms, guided by their fitness and the combination of different biological operators, such as reproduction and mutation.

The Multi-objective Evolutionary Algorithm (MOEA) class is composed of evolutionary algorithms that optimize multi-objective problems (KALYANMOY; DEB, 2001). Evolutionary algorithms are interesting for multi-objective optimization due to their use of populations of solutions. As the solution of multi-objective problems is a Pareto set, composed of multiple non-dominated solutions, it is natural to employ evolutionary algorithms. Instead of selecting the best solution from the population, the population itself could be defined as a solution.

An example of MOEA is the Non-dominated Sorting Genetic Algorithm 2 (NSGA-II) (DEB et al., 2002). This algorithm is a variation of the genetic algorithm, in which the selection and elitism mechanisms were modified to work with multi-objective problems. These modifications are the non-dominated sorting, to sort the best solutions, and the use of crowding distance, to increase diversity (DEB et al., 2002). The algorithm returns the final population as the optimized solution.

Another popular MOEA is the Pareto Archived Evolutionary System (KNOWLES; CORNE, 2000). Individuals are generated using mutations, and the new population is formed using the non-dominance criteria. This method uses the concept of an archive, which is an external set of solutions that is updated each generation. In the case of the PAES, this archive stores the best non-dominated front found in the optimization. This archive is the optimized solution at the end of the algorithm.

To evaluate the quality of a multi-objective, convergence and diversity metrics can be used. Convergence is the efficiency of the algorithm in finding the best solution (EIBEN; SMITH et al., 2003). The convergence metric utilized in this work is the hypervolume metric ($HV$), which can be calculated as (KALYANMOY; DEB, 2001):

$$HV(F) = volume(\bigcup_{i}^{|F|} h_i) \tag{6}$$

where $F$ is a Pareto front, $h_i$ is the hypercube formed by solution $i$ and some reference point $p$. The volume of the hypercube formed by the union of each $h_i$ hypercube gives the space covered by the Pareto set $P$. The greater the metric, the greater is the spread of solutions and the closer it is from the optimal Pareto set (KALYANMOY; DEB, 2001).

Population diversity is a measure of the similarity between solutions of some population (EIBEN; SMITH et al., 2003). The diversity metric utilized in this work is a simple normalized Euclidean distance. This distance is calculated for all pairs of individuals of the population in each generation. Considering this distance, the population diversity can be defined by:

$$Div(P) = \frac{\sum_{x,y \in P} d_{norm}(x,y)}{|P| * (|P|-1)} \tag{7}$$

where $x$ and $y$ are individuals from population $P$, and $d_{norm}$ is the normalized Euclidean distance metric, defined by:

$$d_{norm}(\vec{x},\vec{y}) = \frac{d(\vec{x},\vec{y})}{N} \tag{8}$$

where $d(\vec{x},\vec{y})$ is the Euclidean distance and $N$ is a normalization factor representing the solution space diagonal. This distance is a genotypic diversity metric, that is, it measures the diversity in the problem-independent part of the algorithm.

## 2.6 BIASED RANDOM-KEY GENETIC ALGORITHM

The Biased Random-Key Genetic Algorithm (BRKGA) (GONÇALVES; RESENDE, 2011) is an optimization method of the evolutionary algorithms class. It is a variation of the genetic algorithm in which only the selection and crossover routines are used to evolve solutions (GONÇALVES; RESENDE, 2011).

Initially, all individuals are initialized uniformly at random. Each generation, pairs of individuals are selected and combined through a biased uniform crossover operator, where one parent is an elite solution, and the other is a non-elite. A new population is formed by a fraction of the most-fit solutions of the current generation, a fraction of randomly and uniformly generated solutions, and the remaining individuals are the offspring generated through the crossover operation. In this algorithm, there is no explicit mutation operator, but an implicit mutation can be simulated through the crossover of a random individual with a non-random individual.

One main characteristic of this algorithm is the clear separation between the problem-dependent and independent parts (GONÇALVES; RESENDE, 2011). With this, it is possible to develop problem-specific methods by changing only the method's problem-dependent part. In the BRKGA, this separation occurs in the solution coding, where a decoding function is used to transform a problem-independent codification into a problem solution.

The problem-independent codification is composed of real numbers in the $[0,1]$ interval, decoded into a problem-specific solution using a decoding function, and then evaluated by a fitness function. This way, the algorithm's main structure is standardized for all problems, with only the decoding and fitness functions needing to be developed. This standardization simplifies the algorithm structure by removing the necessity of developing complex solution encodings and various genetic operators, such as crossover or mutation operators.

Similar to other evolutionary algorithms, the BRKGA has a set of parameters that need to be defined. These parameters are the number of iterations ($N_{it}$), population size ($p$), elite fraction ($p_e$), random individuals fraction ($p_m$), and crossover probability ($c_{pr}$). Due to the standardized structure of the algorithm, the crossover operator is fixed.

## 2.7 PARAMETER CONTROL

One of the main aspects of meta-heuristics is the set of parameters that must be tuned. Being generalized optimization frameworks, meta-heuristics can be applied to many types of different problems. For this to happen, however, several parameters must be defined. These values can vary not only for each problem type but can also vary for different instances of the same problem (EIBEN; SMITH et al., 2003) (e.g. a large instance may have different optimal parameter values than a small instance).

Considering evolutionary algorithms, the parameters are generally numerical values, such as population size and crossover probability, and from genetic operators, such as the crossover and mutation probabilities. The BRKGA uses the parameters defined in Section 2.6.

These parameters are usually defined empirically and are fixed through the entire optimization. However, it makes sense that these parameters should be dynamic, using information derived from the optimization process to control and guide the search. Using these dynamic parameters, it is possible to reduce the number of parameters that must be defined by the user and increase the quality of the search itself by properly guiding the optimization (EIBEN; SMITH et al., 2003).

There are several types of parameter control in evolutionary algorithms (PARPINELLI et al., 2019). In this work, an adaptive online parameter control by simple rules is utilized. This technique employs conditional rules that use information from the optimization process, such as fitness and diversity, to modify the parameters during the search.

## 2.8 DECISION-MAKING

In multi-objective optimization, instead of a single solution, we have as output a set of non-dominated solutions. However, most problems still expect only a single solution as final. The selection of a final solution, known as decision-making, is an important last step in multi-objective optimization.

It is possible to select a solution from the Pareto set without using any problem-specific information. One of these generic methods is the *knee* method (BRANKE et al., 2004), in which the most interesting solutions are those where a small improvement in an objective will cause a large deterioration of the others. By focusing only on the knees, the final Pareto set can be greatly reduced. This method, however, does not consider the problem that is being optimized.

For the PSP problem, clustering methods are interesting for decision-making. As Cutello, Narzisi e Nicosia (2006) pointed, finding the native structure can be seen as finding an ensemble of equivalent structures. By grouping similar protein structures, it may be possible to find the native ensemble easier.

Clustering methods can be used to identify these ensembles in the final Pareto set. As the optimization will converge to some point in the search, most of the solutions will have some degree of similarity. Assuming that the optimization model is adequate, it is expected that most solutions will converge to a near-native structure. It can be expected that the native structure will be in the cluster with the largest size.

MUFOLD-CL (ZHANG; XU, 2013) is a fast clustering method explicitly created for protein structure clustering. In this method, protein structures are represented using a distance vector, which contains the pairwise $\alpha$-carbon distance of the structure. More formally, a protein structure can be represented by $D = [d_{11}, d_{12}, d_{13}, ..., d_{1L}, ..., d_{ij}, ..., d_{LL}]$, where $d_{ij}$ is the Euclidean distance between the $i-th$ and $j-th$ $\alpha$-carbon atoms.

To perform the clustering, this method uses two novel protein structure metrics, Dscore1 and Dscore2:

$$\mathrm{Dscore1}(\mathrm{D}^1,\mathrm{D}^2) = \sqrt{\frac{1-\mathrm{dot}\left(\frac{D^1}{\|D^1\|},\frac{D^2}{\|D^2\|}\right)}{2}} \tag{9}$$

$$\mathrm{Dscore2}(\mathrm{D}^1, \mathrm{D}^2) = \frac{1}{L^2} \left( \sum_{1 \leq i,j \leq L} \frac{1}{1 + \left( \frac{d_{ij}^1 - d_{ij}^2}{d_0} \right)^2} \right) \tag{10}$$

In Equation 9, $dot(D^1, D^2)$ is the dot product between vectors $D^1$ and $D^2$; $||D^1||$ and $||D^2||$ are the norms of $D^1$ and $D^2$. In Equation 10, $L$ is the length of the amino acid sequence and $d_0 = 1.24 * \sqrt[3]{L-15} - 1.8$ is a scaling constant defined by the length of the amino acid sequence.

The Dscore1 is a structural difference metric created to be similar to the RMSD score function, while the Dscore2 is a structural similarity metric created to be similar to the Template Modeling score (TM-score) function. While the RMSD and TM-score can be used to create clusters, they are both inefficient for large cluster sizes (ZHANG; XU, 2013). The Dscore1 and Dscore2 metrics were developed to be faster alternatives without losing clustering accuracy (ZHANG; XU, 2013).

The method works by first estimating potential cluster representatives (center of a cluster) using the Dscore1 metric. These representatives are then used to cluster the remaining structures, also using the Dscore1. Finally, after all clusters are formed, new representatives are selected for each cluster using the Dscore2 metric, as this metric is able to describe more accurately the center of a cluster (ZHANG; XU, 2013).

# 3 RELATED WORKS

There are many works in the literature for the PSP problem. Regarding the *de novo* paradigm, the most common approach is with single-objective optimization. However, the multi-objective approach has gained attention in the last decade, with several works proposing multi-objective models and solutions. In this chapter, these works will be presented as they are related to this thesis.

One of the initial works in the area of multi-objective PSP was by Cutello, Narzisi e Nicosia (2006). In this pioneering study, the authors proposed a multi-objective model for the PSP problem, decomposing the CHARMM energy function into two objectives: bonded and non-bonded energies. They demonstrated that these energies were in conflict, and as such, combining them in a single objective would harm the optimization. To optimize the model, they used the IPAES algorithm and also employed a decision-making step to select a final structure by using the method of *knees*.

Handl, Lovell e Knowles (2008) also showed the importance of approaching the PSP problem with a multi-objective model. The authors discussed the characteristics of bonded and non-bonded forces. Hill climbing algorithms were utilized to optimize a single-objective and a multi-objective model of the PSP problem to demonstrate the impact of the multi-objectivization of the problem. For the multi-objective model, the authors decomposed the AMBER energy function into bonded and non-bonded terms.

Tudela e Lopera (2009) proposed a parallel method for the PSP problem with multi-objective model. They also decomposed the CHARMM energy function into bonded and non-bonded terms, further decomposing the non-bonded term into two parts. The final optimization model, with three objectives, was optimized using a parallel NSGA-II algorithm.

Brasil, Delbem e Silva (2013) proposed the use of a multi-objective evolutionary algorithm with many tables (MEAMT) for the PSP problem. In this work, a pure *ab initio* method was proposed, that is, without the use of any other information besides the energy function. The force field CHARMM was used, and the model is composed of four objectives: *Van der Waals*, electrostatic, solvent, and hydrogen bond terms.

Olson e Shehu (2013) proposed a multi-objective evolutionary algorithm (MOEA) for the PSP problem. The author employed the Rosetta framework, using their centroid model, fragment library, and score functions. Regarding the multi-objective model, the Rosetta score4 energy function into three objectives: short-range hydrogen bonding, long-range hydrogen bonding, and a term summing the rest of the energy function.

Faccioli, Bortot e Delbem (2014) proposed the use of the NSGA-II method to optimize the PSP problem. The authors used the CHARMM force field and compared combinations of two and three objectives from a set of objectives. This set included energy function, solvent, hydrogen bonding, compactness, and secondary structure.

Rocha et al. (2015) extended the GAPF framework with phenotypic crowding to optimize the PSP problem. The authors used the GROMOS force field and a multi-objective model with three objectives: energy function, hydrogen bonding, and compactness.

Venske et al. (2016) proposed the ADEMO/D method to optimize the PSP problem. The authors combined an adaptive differential evolution algorithm with the MOEA/D decomposition framework. The CHARMM force field was decomposed into bonded and non-bonded objectives. To select a final structure, the authors compared the decision-making approaches: *knees* method, empiric point, and total energy.

Gao et al. (2017) proposed the use of MOEA to optimize the PSP problem. The authors proposed a multi-objective model composed of the bonded and non-bonded energy from the CHARMM force field. An objective for the solvent-accessible surface area was also considered. A decision-making step was employed using a hierarchical clustering method.

Song et al. (2018a) proposed the use of archive information assisted MOEA (AIMOES) to optimize the PSP problem. The CHARMM force field was decomposed into bonded and non-bonded objectives, and the solvent-accessible surface area was used as the third objective. A hierarchical clustering method was used as a decision-maker.

Song et al. (2018b) proposed the use of a multi-objective PSO to optimize the PSP problem. The authors considered two types of energy functions: physics-based and knowledge-based. The multi-objective model was composed of three objectives, combining both energy functions. The CHARMM force field was selected as physics-based energy and was decomposed into bonded and non-bonded terms, while the DFIRE was selected as knowledge energy and used as a single objective. The MUFOLD-CL clustering method was used as a decision-maker to select a final structure.

Rocha et al. (2018) extended the GAPF framework to consider contact map information as a new objective. The multi-objective model was composed of three objectives, and they were formed by grouping four different terms: energy function, hydrogen bonding, compactness, and contact map. This grouping was determined with an Aggregation Tree method.

Zaman, Parthasarathy e Shehu (2019) proposed the use of MOEA to optimize the PSP problem. The Rosetta energy function score4 was decomposed into three terms: short-range hydrogen bonding, long-range hydrogen bonding, and the rest of the energy function as a single term. Contact map information was considered a separated term, and all four terms were grouped into optimization models with two and four objectives.

Corrêa e Dorn (2019) proposed the use of the ABC method to optimize to PSP problem. Multi-objective models with 2 and 4 objectives were compared, considering the objectives: Rosetta talaris2014 energy function, solvent-accessible surface area, secondary structure, and contact map.

Narloch, Krause e Dorn (2020) compared the algorithm NSGA-II, GDE3, and DEMO in the PSP problem optimization. The authors decomposed the Rosetta score3 energy function into bonded and non-bonded objectives. Secondary structure information was added to the bonded objective.

Chen et al. (2020) proposed the use of multi-objective differential evolution to optimize the PSP problem. The authors proposed the use of the knowledge-based energy function RW-plus, which was decomposed into two objectives: distance energy and orientation energy. The MUFOLD-CL clustering method was used as a decision-maker.

An overview of these works can be seen in Table 2. All of the presented works employ some multi-objective optimization for the PSP problem and represent protein structures with torsion angles. Most of the differences between these works are the algorithms used and the organization of different protein structure information into multiple objectives.

From Table 2, each work is described by their structural model, optimization objectives, protein structure information used, algorithm, and decision-maker. It is possible to see that most of these works use an all-atom model and the division of the energy function into bonded and non-bonded energies. Also, almost all works use evolutionary algorithms.

Table 2 – Related works

| Work | Protein Structure Model | Protein Structure Information | Optimization Objectives | Algorithm | Decision-maker |
|---|---|---|---|---|---|
| Cutello, Narzisi e Nicosia (2006) | all atoms | CHARMM force field<br>secondary structures<br>super secondary structures<br>rotamers | bonded<br>non-bonded | IPAES | knees |
| Handl, Lovell e Knowles (2008) | all atoms | AMBER force field | bonded<br>non-bonded | hill climbing | - |
| Tudela e Lopera (2009) | all atoms | CHARMM force field<br>secondary structures<br>super secondary structures<br>rotamers | bonded<br>van der Waals<br>non-bonded | NSGA-II | knees |
| Brasil, Delbem e Silva (2013) | all atoms | CHARMM force field | van der Waals<br>electrostatic<br>solvent<br>hydrogen bonding | MEAMT | - |
| Olson e Shehu (2013) | centroid | Rosetta force field<br>fragments | score4 energy<br>short hydrogen bond<br>long hydrogen bond | MOEA | - |

| | | | | | |
|---|---|---|---|---|---|
| Faccioli, Bortot e Delbem (2014) | all atoms | CHARMM force field<br>secondary structures<br>rotamers<br>CADB-2 database | energy function<br>solvent<br>hydrogen bond<br>compactness<br>secondary structure | NSGA-II | - |
| Rocha et al. (2015) | centroid | GROMOS force field<br>secondary structure | energy function<br>hydrogen bond<br>compactness | GA | - |
| Venske et al. (2016) | all atoms | CHARMM force field<br>secondary structure | bonded<br>non-bonded | ADEMO/D | knees<br>empirical point<br>total energy |
| Gao et al. (2017) | all atoms | CHARMM force field<br>secondary structure<br>rotamers | bonded<br>non-bonded<br>solvent | MOEA | clustering |
| Song et al. (2018a) | all atoms | CHARMM force field<br>DFIRE force field<br>secondary structure<br>rotamers | bonded<br>non-bonded<br>DFIRE | MOPSO | clustering |
| Song et al. (2018b) | all atoms | CHARMM force field<br>secondary structure<br>rotamers | bonded<br>non-bonded<br>solvent | AIMOES | clustering |

| | | | | | |
|---|---|---|---|---|---|
| Rocha et al. (2018) | centroid | GROMOS force field<br>secondary structure<br>fragments<br>contact map | energy function<br>hydrogen bond<br>compactness<br>contact map | GA | - |
| Zaman, Parthasarathy e Shehu (2019) | centroid | Rosetta force field<br>fragments<br>contact map | score4 energy<br>short hydrogen bond<br>long hydrogen bond<br>contact map | MOEA | - |
| Corrêa e Dorn (2019) | all atoms | Rosetta force field<br>secondary structure<br>contact map<br>angle probability list | talaris2014<br>secondary structure<br>contact map<br>solvent | ABC | - |
| Narloch, Krause e Dorn (2020) | centroid | Rosetta force field<br>secondary structure<br>angle probability list | bonded<br>non-bonded | NSGA-II<br>DEMO<br>GDE3 | - |
| Chen et al. (2020) | all atoms | RWplus force field<br>secondary structure<br>rotamers | distance energy<br>orientation energy | DE | clustering |
| **This work** | **centroid** | **Rosetta force field**<br>**secondary structure**<br>**fragments**<br>**contact map** | **score4 energy**<br>**secondary structure**<br>**contact map** | **BRKGA** | **clustering** |

Source: Author

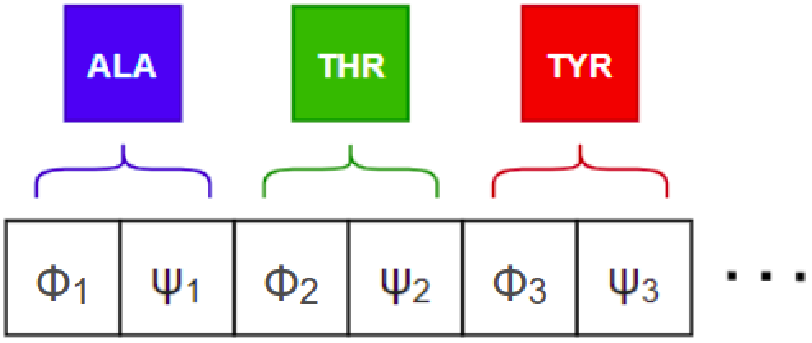
# 4 PROPOSED METHOD

In this work, an *de novo* method is proposed to approach the PSP problem. To this end, a multi-objective model was proposed, and a multi-objective BRKGA was developed to optimize the model. The MUFOLD-CL clustering method was employed to select the final structure from the optimized Pareto set.

## 4.1 PROTEIN REPRESENTATION

Proteins are modeled with the Rosetta framework using the $C_\alpha$ backbone representation (ROHL et al., 2004), where the side chain is simplified as a centroid. The $\phi$, $\psi$, and $\omega$ angles are used to model each amino acid. These angles are in the $[-180, 180]$ domain, except for the $\omega$ angle, whose optimal value is always 180º (CUTELLO; NARZISI; NICOSIA, 2006). Figure 8 shows how an amino acid sequence can be coded as a list of angles.

Figure 8 – Protein structure representation as a list of torsion angles. In this example, the first three amino acids are alanine (ALA), threonine (THR), and tyrosine (TYR). Each amino acid has a $\phi$ and $\psi$ angle. As the $\omega$ angle is fixed, it is omitted from the codification.



Source: Author

## 4.2 MATHEMATICAL MODEL

The proposed model for the PSP problem is composed of 3 objectives, shown in Expressions 11-13:

$$\min \quad score4(\vec{x}) \tag{11}$$

$$\max \quad SS(\vec{x}) \tag{12}$$

$$\max \quad CM(\vec{x}) \tag{13}$$

The $\vec{x}$ vector represents a protein structure coded as a list of angles, as described in Section 4.1, with size $2L$, where $L$ is the length of the amino acid sequence of the protein. The angle list is mapped into a protein structure before being evaluated by each objective function. Expression 11 is the *score4* energy function from the Rosetta framework (ROHL et al., 2004) and is used to evaluate the protein structure.

Expression 12 refers to secondary structure information predicted by the PSIPRED server. This prediction was performed excluding homologous proteins. Equation 14 details the evaluation, where for each amino acid $x_i$, the $pSS$ function will return the predicted probability for the current secondary structure manifested, determined with the DSSP program (KABSCH; SANDER, 1983), which assigns secondary structure to proteins structures. Hence, protein structures with the most probable secondary structures are benefited.

$$SS(\vec{x}) = \sum_{i}^{L} pSS(x_i) \tag{14}$$

Expression 13 refers to contact map information predicted by the RaptorX server. This prediction was performed removing homologous proteins. Two atoms are considered to be in contact if their distance is less than the cutoff distance. Equation 15 describes how each contact is evaluated for a given protein structure.

For each contact $c_i$ in the $L$-best list, the pair of amino acids $c_i^1$ and $c_i^2$ from this contact is verified. If their distance, $d(c_i^1, c_i^2)$, is less than the cutoff value (8Å), the pair is said to be in contact, and a term equal to the predicted probability of this contact is summed. Otherwise, this probability term is decreased exponentially due to the cutoff value's distance, allowing slight deviations from the cutoff to be considered. By using this objective function, protein structures that exhibit the most probable contacts are benefited.

$$CM(\vec{x}) = \sum_{i}^{L} \begin{cases} p_i & \text{if} \quad d(c_i^1, c_i^2) \leq 8 \\ \frac{p_i}{e^{d(c_i^1, c_i^2) - 8}} & \text{otherwise} \end{cases} \tag{15}$$

## 4.3 MULTI-OBJECTIVE BRKGA

A multi-objective BRKGA was developed, named MO-BRKGA. The proposed algorithm uses the base structure of the original BRKGA (GONÇALVES; RESENDE, 2011) and can be applied to any multi-objective problem that can be approached with evolutionary algorithms. One of the main characteristics of the BRKGA is that there is a clear division between the problem-independent part, which generates and modifies individuals, and the problem-dependent part, which decodes and evaluates individuals. As such, few modifications are needed to develop a problem-specific optimizer.

The main modifications were to allow the algorithm to optimize multi-objective problems. Modifications were made to the problem-independent parts to achieve multi-objective optimization. The elitist selection operator from NSGA-II (DEB et al., 2002) was used to generate the elite part of the population.

This selection divides the population into non-dominated sets (DEB et al., 2002). These sets are sorted by dominance, with the first set not dominated by any solution, the second set dominated by the first, etc. Inside each non-dominated set, the solutions are further sorted by their crowding distance, which measures the proximity of solutions (DEB et al., 2002). The solutions are selected first by adding the sorted solution sets, and if the addition of this set extrapolates the maximum size, single solutions are added considering the sorting.

An archive (set of solutions) is maintained and updated at each generation. This archive has the same size as the population, and it is updated using the non-dominated sorting with crowding. At the end of the algorithm, the archive is returned and contains the best Pareto set found. However, it may also contain dominated solutions as the best Pareto set may not occupy the entire archive.

Maintaining dominated solutions is interesting as some problems may benefit from having sub-optimal solutions. For example, the decision-making method applied in this work is a clustering method, and due to this, having more solutions will help the formation of clusters.

### 4.3.1 Parameter control

In this work, an adaptive online parameter control by simple rules is employed to control and guide the optimizer. The information utilized to guide the search is the population diversity defined in Equation 7 in Section 2.5. This work will not incorporate fitness information in the parameter control, as the definition of fitness is different for a multi-objective optimizer, and concepts such as best and mean fitness are not trivially defined.

To incorporate the use of diversity information, the algorithm is modified to control and increase diversity in the population. In the original BRKGA method, the crossover probability $p_{cr}$ is defined as a parameter of the algorithm and should have a value in the range $(0.5, 1)$. This range of values characterizes the bias towards the elite solution as these solutions will always have a greater chance of passing their values. However, this parameter is static through the entire optimization, and the use of higher values will decrease the diversity as the generated solutions will inherit most of the elite parents. In this work, the crossover parameter is modified to be a uniformly random value in the range $(0.5, 1)$, removing the complexity of defining an optimal value for this parameter and minimizing the impact in the diversity. This value is generated for each crossover operation.

Another modification is the introduction of two diversity control values: the diversity fraction ($p_d$) and the exploration diversity threshold ($\delta$). These values are used to control the generation of the elite part of the population. The diversity fraction indicates the fraction of the elite part that should be diversified, while the exploration diversity threshold is used to indicate the minimum distance between two diversified solutions during the exploration part of the search.

The set $P_d$ of diversified elite solutions is defined as:

$$P_d = \{\forall \vec{x} \in P_d \mid \Delta(\vec{x}) \geq \delta\} \tag{16}$$

where $\Delta(\vec{x})$ is defined as:

$$\Delta(\vec{x}) = \min_{\substack{\vec{y} \in P_d \\ \vec{x} \neq \vec{y}}} d_{norm}(\vec{x}, \vec{y}) \tag{17}$$

where $d_{norm}$ is the normalized Euclidean distance expressed by Equation 8. This definition of diversified individuals spreads the solutions through the space. Incorporating this diversification in the elite part allows the algorithm to explore the best solutions in different parts of the search space.

Given the population size $p$, $E = p \times p_e$ is the size of the elite part of the population and $E_d = E \times p_d$ is the size of the diversified elite. With $P_s$ as the sorted population, the elite part $P_e$ is generated by the following steps:

- While $|P_e| < E$, do:
  1. If $|P_d| < E_d$, then:
     a) If exists $\vec{x}_i \in P_s$ that can be inserted in $P_d$, then select $\vec{x}_i$ with lowest index $i$.
     b) Else, from $P_s$ select $\vec{x}_i$ with greatest $\Delta$
     c) Insert selected $\vec{x}_i$ in $P_d$ and $P_e$
     d) Remove selected $\vec{x}_i$ from $P_s$
  2. Else:
     a) From $P_s$ select $\vec{x}_i$ with lowest index $i$
     b) Insert selected $\vec{x}_i$ in $P_e$
     c) Remove selected $\vec{x}_i$ from $P_s$

The parameter control is performed by dividing the optimization procedure into two phases: exploration and exploitation. In the exploration phase, the algorithm generates diversified solutions to search the entire solution space. In the exploitation phase, the algorithm increases the evolutionary pressure by favoring the best solutions found. By properly controlling the algorithm parameters, it is possible to balance global and local searches.

To balance between exploration and exploitation, each phase runs for $N_{it}/2$ iterations. For both phases, the parameters $p_e$ and $p_m$ are initially set to 0.25. With these values, half of the population is composed of offspring. With $p_e = 0.25$, it also means that each elite individual should on average generate two offspring each generation.

This initial value was defined empirically. The following initial values and ranges for each parameter were also defined empirically. Although these values may not always be optimal, they are simple and reasonable enough to be used as base values.

#### *4.3.1.1 Exploration*

The exploration phase initializes the optimization process. The diversification parameters $p_d$ and $\delta$ are used to explore the search space. The diversification fractions starts as $p_d = 0$, and is modified in steps of $k = 0.01$. This parameter is updated during the exploration phase to keep the population diversity $Div(P)$ above $\delta$. If the current $Div(P) < \delta$, $p_d$ is increased by $k$. Otherwise, $p_d$ is decreased by $k$. The parameter $p_d$ is kept in the range $[0, 0.5]$, diversifying at most half of the elite part.

If the $Div(P)$ is still below $\delta$ when $p_d = 0.5$, then the parameter $p_m$ is also modified. The parameter is kept in the range $[0.25, 0.5]$ and is updated using step $k$. By increasing the random part of the population, the diversity of the population is increased. At the maximum value of $p_m = 0.5$, half of the population is uniformly randomly generated each iteration. Also, only $1/4$ of the population is offspring, with each elite individual generating on average one offspring and each non-elite parent being on average $2/3$ of the time a random individual.

The exploration diversity threshold $\delta$ is used to define the diversity of the exploration phase. It can be defined by the user as an algorithm parameter. However, the value 0.4 should be reasonable, in general, and is used as the default value for this parameter. This value was selected due to the nature of uniformly randomly generated solutions and global search.

It is important to search all the space evenly in generalized problem optimization where there is no information about the solution space. To this end, the uniform distribution is interesting for the generation of solutions. Random solutions generated using this distribution are evenly dispersed in the solution space. For a sufficiently large population of uniformly random individuals, it is easy to see that the average normalized Euclidean distance $d_{norm}$ is close to 0.5.

Considering this, $\delta$ is set to 0.4 by default. With this value, the population of the exploration phase has the property of being similar to a uniform distribution without losing the generation of good solutions. Although, in general, it should not be necessary to change this value, in some cases, it can be interesting to lower the parameter to limit the exploration.

#### *4.3.1.2 Exploitation*

In the exploitation phase, the algorithm eliminates the diversity parameters. The parameter $p_m$ is reset to the initial value 0.25 if it was modified during the exploration. The only parameter modified during this phase is the elite fraction $p_e$.

During this phase, the algorithm forces the search to converge to the best solution found. To increase the convergence, the parameter $p_e$ is increased in the range $[0.25, 0.5]$, using step $k$. At the maximum value of $p_e = 0.5$, half of the population is composed of elite solutions, and each elite individual will generate, on average, one offspring.

As the algorithm converges, the diversity is decreased to some lower value. In single-objective optimization, this value should be close to zero, indicating that the population is composed of the best solution and individuals similar to it. However, in multi-objective optimization, there are multiple solutions in the optimal frontier. The distribution of these solutions is different for each problem. Therefore the final population diversity may not be a reasonable convergence indicator.

#### *4.3.1.3 Parameter removal*

Considering the modifications proposed, the parameters $p_e$, $p_m$ and $c_{pr}$ are no longer user-defined. The only parameters that still need to be defined are the number of iterations $N_{it}$ and the population size $p$. The exploration diversity threshold $\delta$ can also be defined if necessary.

This proposed parameter control aims to find a reasonable balance between usability complexity and optimization efficiency. The proposed control does not guarantee the selection of optimal values for every problem. However, it should select values that are reasonable for any general optimization, as it uses the generic concept of exploration and exploitation. Although the algorithm may not execute the most optimal search for some specific problem, the decreased number of parameters allows the user to focus more on the problem modeling and less on the calibration of the algorithm.

### 4.3.2 Parallelism

To increase the scalability of the algorithm, a master-slave model was developed using CPU threads. During fitness evaluation, the master thread divides the population into equal-sized chunks and distributes them to other threads. More formally, considering a number $t$ of threads (e.g. the number of processors), the master will divide the population into chunks of size $p/t$. Each thread will evaluate the fitness of individuals in its chunk, effectively reducing the processing time of fitness evaluation, which is usually the most time-consuming step of the optimizer.

### 4.3.3 Codification and fitness

All the previous changes were in the problem-independent part of the algorithm. To be able to optimize some given problem, it is also necessary to define the problem-dependent part. In the BRKGA, this part consists of defining the decoder function and the fitness function.

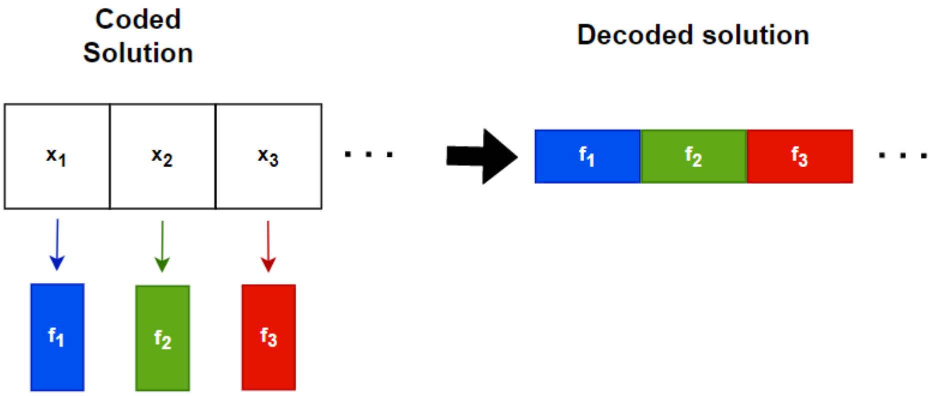
The fitness for the multi-objective model of the PSP is a tuple $(F_1, F_2, F_3)$, where $F_1$, $F_2$, and $F_3$ are the objectives defined by the Expressions 11-13, respectively. The decoder function is a function that will receive a coded solution $x = (x_1, x_2, x_3, ..., x_n)$, where $n$ is the size of an encoded chromosome, which is problem-specific, and each $x_i$ is in the domain $[0, 1]$. The decoder should map this chromosome into a problem-specific solution, which will then be evaluated by the fitness function.

In this work, two decoders are used in two different executions of the algorithm. The first decoder, named fragment decoder, takes an $F = L/9$-sized chromosome, with $L$ being the amino acid sequence length, and maps it into an angle list (defined in Section 4.1) by inserting fragments of size nine. Each of these fragments is a continuous sequence of nine amino acids extracted from some known protein structure.

For each amino acid in the protein to be predicted, 200 fragments of size nine were generated using the Robetta server. These fragments were predicted excluding homologous proteins. The torsion angles $\phi$ and $\psi$ are extracted from each fragment and used to build the solution. As the fragments have size nine, each selected fragment contributes with 9 pairs of $\phi$ and $\psi$ angles in the solution.

In the fragment decoder, each variable of the chromosome $x$ (candidate solution) is used to select a fragment from the list of 200 fragments of an amino acid. The $x_i$ variable represents a fragment $f_i$ that starts on the amino acid with position $p_i = 9 \times (i-1)+1$ in the protein sequence. The inserted fragment $f_i$ is the one with position $\lceil 200 \times x_i \rceil$ in the fragment list of amino acid $p_i$. Figure 9 shows the decoding process.

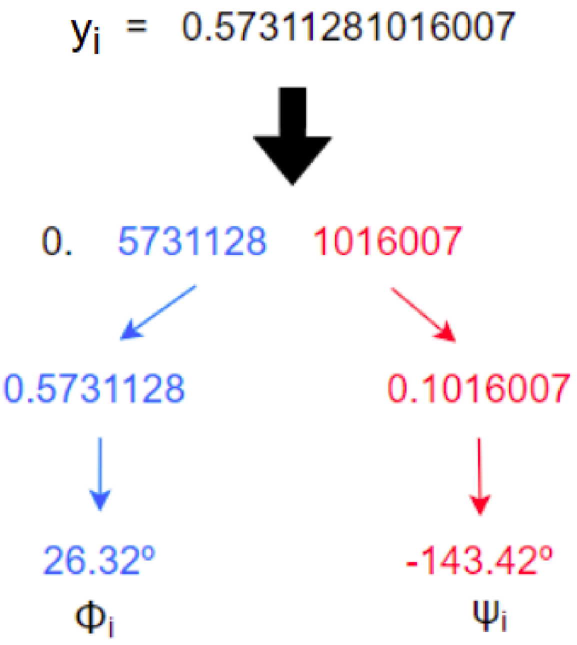
Figure 9 – Fragment decoder. Each value of the individual $x$ is a real number $x_i \in [0,1]$ that is mapped to a integer $f_i \in [1,200]$. This integer represents the index of some fragment of size nine whose residue angles will be inserted in the decoded solution.



Source: Author

The second decoder, named residue decoder, takes an $L$-sized chromosome $y$ and maps it into an angle list by extracting the $\phi$ and $\psi$ angles from each variable. The $y_i$ variable is mapped into torsion angles $\phi_i$ and $\psi_i$ by using 14 digits as scale, where the first 7 digits are used to generate $\phi_i$ and the remaining 7 are used to generate $\psi_i$. If $D_i = \{d_1, d_2, ..., d_7\}$ are the 7 digits used to generate $\phi_i$, the angle $\phi_i$ can be defined as $\phi_i = -180 + 360 \times r$ where $r = 0.d_1 d_2 ... d_7$ (the $\psi$ angle is defined similarly). Figure 10 exemplifies this decoding process.

Figure 10 – Residue decoder. In this example, a single chromosome value, $y_i$, is pictured. From this value is extracted a single pair of angles $\phi$ and $\psi$. This procedure happens for all chromosome values.



Source: Author

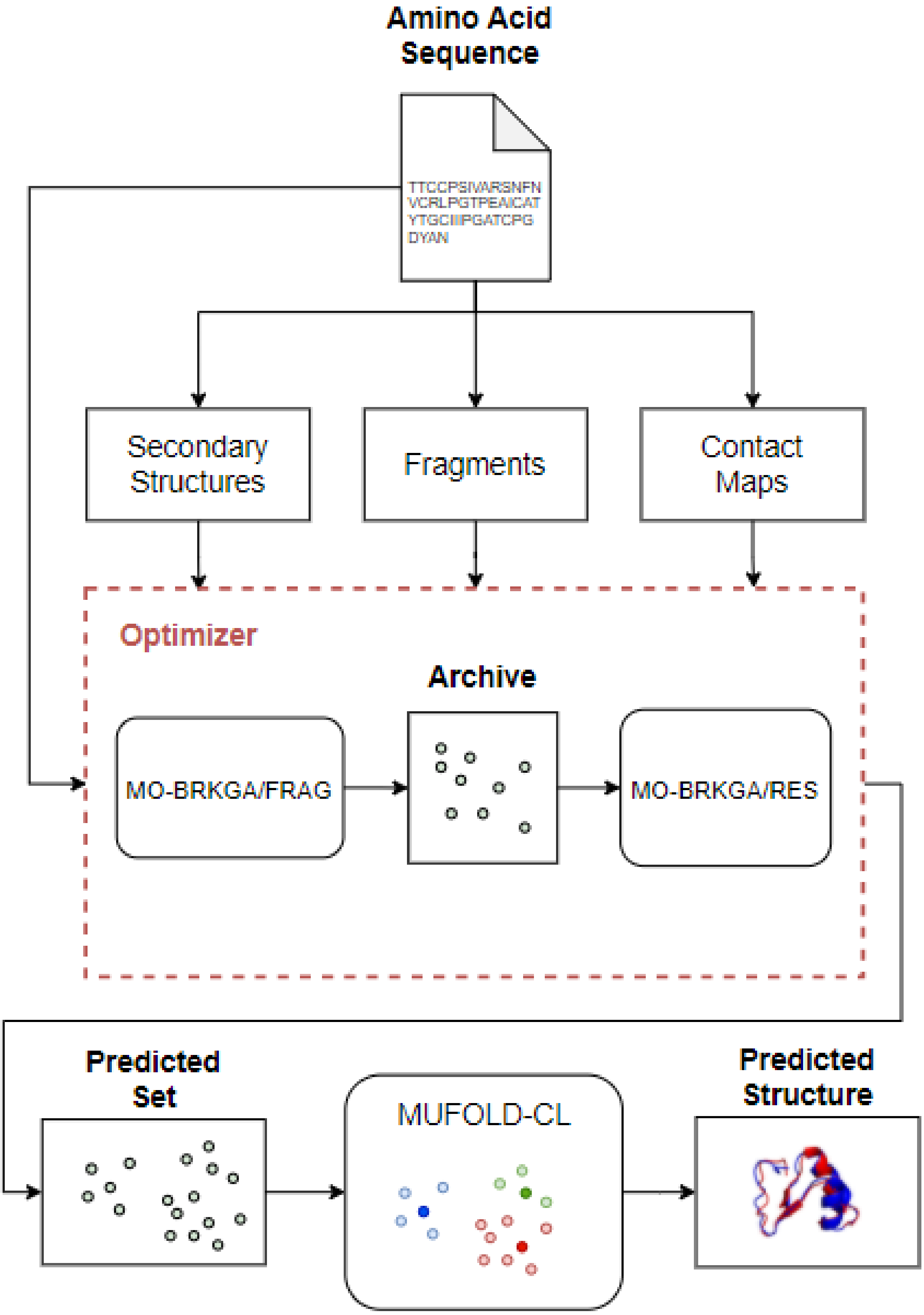
## 4.4 PREDICTOR

The proposed model first applies the MO-BRKGA with the fragment decoder, named MO-BRKGA/FRAG, to predict protein structures. This algorithm will search the structure space using fragments, generating valid low-resolution structures. Then, the MO-BRKGA with the residue decoder, named MO-BRKGA/RES, optimizes the archive of solutions returned by the MO-BRKGA/FRAG.

Using the archive from MO-BRKGA/FRAG as the initial population, the MO-BRKGA/RES can refine the results and increase their resolution. The archive returned by the MO-BRKGA/RES is the predicted set of protein structures. A decision-making step is applied to select the final predicted protein structure. To accomplish that, the MUFOLD-CL method (ZHANG; XU, 2013) is used to cluster the final set, and the centroid of the largest group (determined by MUFOLD-CL) is returned as the predicted structure of the proposed method.

A diagram with a visualization of the full predictor can be seen in Figure 11. In this diagram, the main input is the amino acid sequence. From this sequence, the secondary structure, contact map, and fragment information are generated. This information is used to feed the optimizer, which starts with MO-BRKGA/FRAG method. This method generates an archive of solutions, used as the initial population for the second method, MO-BRKGA/RES. The output of the second method is the optimizer output. This output is also an archive of solutions, used as input for the clustering method MUFOLD-CL. The final structure is then selected from the largest cluster found.

Figure 11 – Proposed predictor



Source: Author

# 5 EXPERIMENTS, RESULTS, AND ANALYSIS

## 5.1 EXPERIMENTATION PROTOCOL

Different tests are performed to analyze the performance of the proposed predictor. These experiments are divided into three scenarios:

- **Processing time analysis**: analysis of the computational performance of the optimizer with regard to the time to predict a single structure;
- **Optimizer performance analysis**: analysis of the optimizer search performance on the search space; and
- **Predictor performance analysis**: analysis of the quality of structures predicted with the proposed method.

Each of these experiment scenarios is conducted using the same algorithm parameters, with the number of iterations $N_{it} = 1000$ and population size $p = 500$. Both MO-BRKGA/FRAG and MO-BRKGA/RES use the same values, resulting in 1,000,000 fitness evaluations. However, the MO-BRKGA/RES also defines the exploration diversity threshold as $\delta = 0.25$. This definition occurs to limit the exploration of new solutions in the MO-BRKGA/RES, whose objective is to refine the solutions found by the MO-BRKGA/FRAG.

The experiments utilize some subset of the proteins listed in Table 3. All proteins were taken from the RCSB database [1], with the exception of proteins T0868, T0900, T0968s1, and T1010, which were taken from the CASP competition [2].

All the experiments were executed 20 times for statistical validation. The proposed predictor was implemented using C++17, and experiments were performed on an Ubuntu 18.04 system with an Intel Xeon E7-8860 @ 80x 2.26GHz and 1TB RAM. The utilized system has a NUMA architecture with four nodes, each node with 20 cores (10 physical and 10 virtual).

[1] https://www.rcsb.org/

[2] https://predictioncenter.org/

Table 3 – Proteins utilized in the experiments

| **Protein** | **Size** | **Class** |
|---|---|---|
| 1ACW | 29 | $\alpha\beta$ |
| 1DFN | 30 | $\beta$ |
| 1ZDD | 34 | $\alpha$ |
| 1I6C | 39 | $\beta$ |
| 2MR9 | 44 | $\alpha$ |
| 2P81 | 44 | $\alpha$ |
| 1AB1 | 46 | $\alpha\beta$ |
| 1CRN | 46 | $\alpha\beta$ |
| 1ENH | 54 | $\alpha$ |
| 1GB1 | 56 | $\alpha\beta$ |
| 2KDL | 56 | $\alpha$ |
| 1BDD | 60 | $\alpha$ |
| 1ROP | 63 | $\alpha$ |
| 1AIL | 73 | $\alpha$ |
| 1HHP | 99 | $\beta$ |
| T0900 | 106 | $\beta$ |
| T0968s1 | 119 | $\alpha\beta$ |
| 1ALY | 146 | $\beta$ |
| T0868 | 161 | $\alpha$ |
| T1010 | 210 | $\beta$ |

Source: Author

## 5.2 RESULTS AND ANALYSIS

### 5.2.1 Online Parameter Control Influence

Before starting the experiments, an initial comparison between the proposed predictor was made with a simpler version of the method. This comparison uses the GDT_TS metric, which is more robust than the RMSD metric, and was made to validate the use of parameter control. Considering the trade-off between the optimizer usability and efficiency, the proposed method should generate results with quality similar or better than the simpler algorithm to be considered useful.

The simplified optimizer is called MO-BRKGA, and the proposed optimizer with parameter control is called MO-BRKGA + PC. The simplified version has the same structure as the proposed predictor, including the diversity modifications, but uses static parameters defined before the algorithm execution. These parameters are listed in Table 4 and were defined empirically. Both MO-BRKGA/FRAG and MO-BRKGA/RES use the same parameters. Both algorithms were executed 20 times for statistical validation.

The results can be seen in Table 5. For each version of the optimizer, the best solution and the solution selected with MUFOLD-CL were considered. The ANOVA test was performed to validate the statistical difference between the result of the proposed predictor MO-BRKGA + PC and the simplified version. The result of the ANOVA test can be seen in Tables 12 and 13 in Appendix A.

The last two rows B/S/W summarize the results at the end of Table 5. B represents the number of proteins where the competing method was statistically better than the proposed predictor, and the W is the number of times where the competing method was statistically worse than the proposed predictor. S is the number of proteins where there was no statistical difference between the results of the competing method and the proposed predictor.

Table 4 – MO-BRKGA parameters

| **Iteration Number**<br>$N_{it}$ | **Population Size**<br>$p$ | **Elite Fraction**<br>$p_e$ |
|---|---|---|
| 1000 | 500 | 0.5 |
| **Mutant Fraction**<br>$p_m$ | **Diversified Fraction**<br>$p_d$ | **Diversity Distance**<br>$\delta$ |
| 0.2 | 0.5 | 0.2 |

Source: Author

Observing the results, it is possible to see that both optimizer versions generate the same best solutions. The first B/S/W row indicates that both MO-BRKGA (best) and MO-BRKGA + PC (best) have statically equivalent results. The second B/S/W row indicates that most of the results between the MO-BRKGA (MUFOLD-CL) and MO-BRKGA + PC (MUFOLD-CL) were statistically equivalent.

Considering these results, it is possible to conclude that both algorithms have similar results. As one of the objectives of the parameter control is to reduce the usability complexity of the algorithm, the MO-BRKGA with parameter control has an advantage over the simpler MO-BRKGA. Therefore, the use of parameter control is validated, and all experiments following this were done using the complete version of the predictor.

Table 5 – Comparison of MO-BRKGA and MO-BRKGA + PC using the GDT_TS metric. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| Protein | | MO-BRKGA<br>Best | MO-BRKGA<br>MUFOLD-CL | MO-BRKGA + PC<br>Best | MO-BRKGA + PC<br>MUFOLD-CL |
|---|---|---|---|---|---|
| 1AB1 | $f^*$ | **81.74** | 78.26 | 78.70 | 70.43 |
| | $\bar{x}$ | 76.91 | 65.28 | 73.22 | 64.33 |
| | $s$ | 2.90 | 8.97 | 3.46 | 3.84 |
| 1ACW | $f^*$ | 68.97 | 58.62 | **70.34** | 60.69 |
| | $\bar{x}$ | 57.31 | 46.07 | 60.79 | 49.86 |
| | $s$ | 6.34 | 5.32 | 4.95 | 7.09 |
| 1AIL | $f^*$ | **60.57** | 41.71 | 51.14 | 40.57 |
| | $\bar{x}$ | 44.34 | 34.26 | 44.47 | 35.11 |
| | $s$ | 5.36 | 3.37 | 3.39 | 3.70 |
| 1ALY | $f^*$ | 28.22 | 23.70 | **28.90** | 19.32 |
| | $\bar{x}$ | 21.43 | 16.23 | 20.40 | 14.55 |
| | $s$ | 3.37 | 3.28 | 3.21 | 2.30 |
| 1BDD | $f^*$ | 74.67 | 71.33 | **76.67** | 73.33 |
| | $\bar{x}$ | 70.55 | 66.15 | 72.52 | 66.68 |
| | $s$ | 2.65 | 2.69 | 2.60 | 2.96 |
| 1CRN | $f^*$ | **88.70** | 86.09 | 84.78 | 77.39 |
| | $\bar{x}$ | 82.43 | 71.46 | 80.91 | 69.33 |
| | $s$ | 2.70 | 10.46 | 2.24 | 6.76 |
| 1DFN | $f^*$ | **80.00** | 67.33 | 78.00 | 72.00 |
| | $\bar{x}$ | 70.33 | 57.13 | 63.37 | 55.10 |
| | $s$ | 5.32 | 5.72 | 6.09 | 8.07 |
| 1ENH | $f^*$ | 85.19 | 78.52 | **85.93** | 73.70 |
| | $\bar{x}$ | 80.44 | 69.52 | 77.39 | 66.69 |
| | $s$ | 2.96 | 5.31 | 3.99 | 5.61 |
| 1GB1 | $f^*$ | **85.00** | 80.71 | 79.29 | 77.14 |
| | $\bar{x}$ | 76.64 | 69.59 | 74.27 | 66.39 |
| | $s$ | 4.13 | 5.79 | 3.71 | 5.66 |
| 1HHP | $f^*$ | **59.39** | 46.46 | 44.85 | 43.23 |
| | $\bar{x}$ | 39.30 | 31.19 | 36.22 | 28.23 |
| | $s$ | 7.96 | 7.76 | 4.93 | 5.97 |
| 1I6C | $f^*$ | **69.23** | 63.08 | 69.23 | 66.15 |
| | $\bar{x}$ | 64.79 | 58.56 | 65.18 | 60.46 |
| | $s$ | 2.55 | 2.61 | 2.83 | 3.01 |
| 1ROP | $f^*$ | **92.50** | 78.57 | 92.14 | 85.00 |
| | $\bar{x}$ | 84.48 | 64.04 | 75.16 | 62.79 |
| | $s$ | 5.31 | 7.44 | 7.11 | 12.72 |
| 1ZDD | $f^*$ | 91.76 | 87.65 | **92.35** | 88.24 |
| | $\bar{x}$ | 88.47 | 84.91 | 88.03 | 83.50 |
| | $s$ | 1.53 | 2.21 | 2.14 | 2.46 |

| | | | | | |
|---|---|---|---|---|---|
| 2KDL | $f^*$ | 46.43 | 41.07 | **48.93** | 38.93 |
| | $\bar{x}$ | 42.36 | 36.20 | 42.58 | 34.89 |
| | $s$ | 1.86 | 2.64 | 2.68 | 2.67 |
| 2MR9 | $f^*$ | **91.82** | 82.73 | 91.82 | 86.36 |
| | $\bar{x}$ | 87.95 | 75.66 | 86.99 | 77.08 |
| | $s$ | 2.33 | 3.14 | 3.17 | 4.87 |
| 2P81 | $f^*$ | **68.64** | 58.18 | 67.27 | 58.18 |
| | $\bar{x}$ | 62.98 | 53.00 | 63.85 | 52.58 |
| | $s$ | 3.38 | 3.73 | 1.94 | 4.02 |
| T0868 | $f^*$ | 60.17 | 51.38 | **62.41** | 48.28 |
| | $\bar{x}$ | 47.02 | 38.87 | 47.09 | 39.50 |
| | $s$ | 6.96 | 6.52 | 7.19 | 4.44 |
| T0900 | $f^*$ | 39.41 | 34.90 | **41.18** | 33.53 |
| | $\bar{x}$ | 30.08 | 22.86 | 31.42 | 23.44 |
| | $s$ | 4.50 | 4.09 | 4.37 | 4.62 |
| T0968S1 | $f^*$ | 51.02 | 42.88 | **52.71** | 46.10 |
| | $\bar{x}$ | 41.09 | 33.05 | 42.60 | 34.73 |
| | $s$ | 4.77 | 5.71 | 5.31 | 6.77 |
| T1010 | $f^*$ | **24.57** | 18.19 | 20.19 | 17.14 |
| | $\bar{x}$ | 18.41 | 14.08 | 17.95 | 13.81 |
| | $s$ | 2.84 | 2.48 | 1.77 | 1.69 |
| **B/S/W MO-BRKGA + PC (best)** | | 0/20/0 | 0/0/20 | - | 0/0/20 |
| **B/S/W MO-BRKGA + PC (MUFOLD-CL)** | | 20/0/0 | 3/15/2 | 20/0/0 | - |

Source: Author

### 5.2.2 Parallel Model Definition and Processing Time Analysis

In this experiment, the execution time of the proposed optimizer is analyzed. The architecture of the computer utilized to execute the tests was also taken into consideration. The computer has 80 cores (40 physical and 40 virtual) in a NUMA architecture with four nodes, each having 20 cores.

Control mechanisms were employed to use this architecture properly. As the parallelism was achieved using the OpenMP [3] 4.5 library, two OpenMP options were used to control the distribution of threads: `OMP_PROC_BIND = close` and `OMP_PLACES = cores`. These options change the thread affinity of the application, which allows the placement of threads in specific cores (DREPPER, 2007).

[3] https://www.openmp.org/

This experiment used four proteins from Table 3 and are displayed in Table 6. These proteins were selected to identify possible impacts of different sizes and classes of proteins in the running time. For each protein, the optimizer was executed 20 times. The average time (with standard deviation) in seconds was measured. The results of the experiment are shown in Table 7 and in Figure 12.

In Table 7, it is possible to see the execution time for each protein considering 1, 2, 4, 6, 8, 10, 20, 40, and 80 threads. The execution time reduction is clear from the serial execution (1 thread) to the best case (10 threads). The use of 10 threads was the best in this experiment due to the computer architecture utilized.

Although the computer has 80 cores, only 40 are physical, and there are only 10 physical cores in each NUMA node. Using multiples NUMA nodes in the application incurs a communication cost between the different nodes, considerably increasing the execution time. This behavior can be seen in the graphs of Figure 12, with a sharp increase between 10 and 40 threads.

Table 6 – Selected proteins for execution time analysis

| **Protein** | **Size** | **Class** |
|---|---|---|
| 1ZDD | 34 | $\alpha$ |
| 1GB1 | 56 | $\alpha\beta$ |
| 1AIL | 73 | $\alpha$ |
| 1HHP | 99 | $\beta$ |

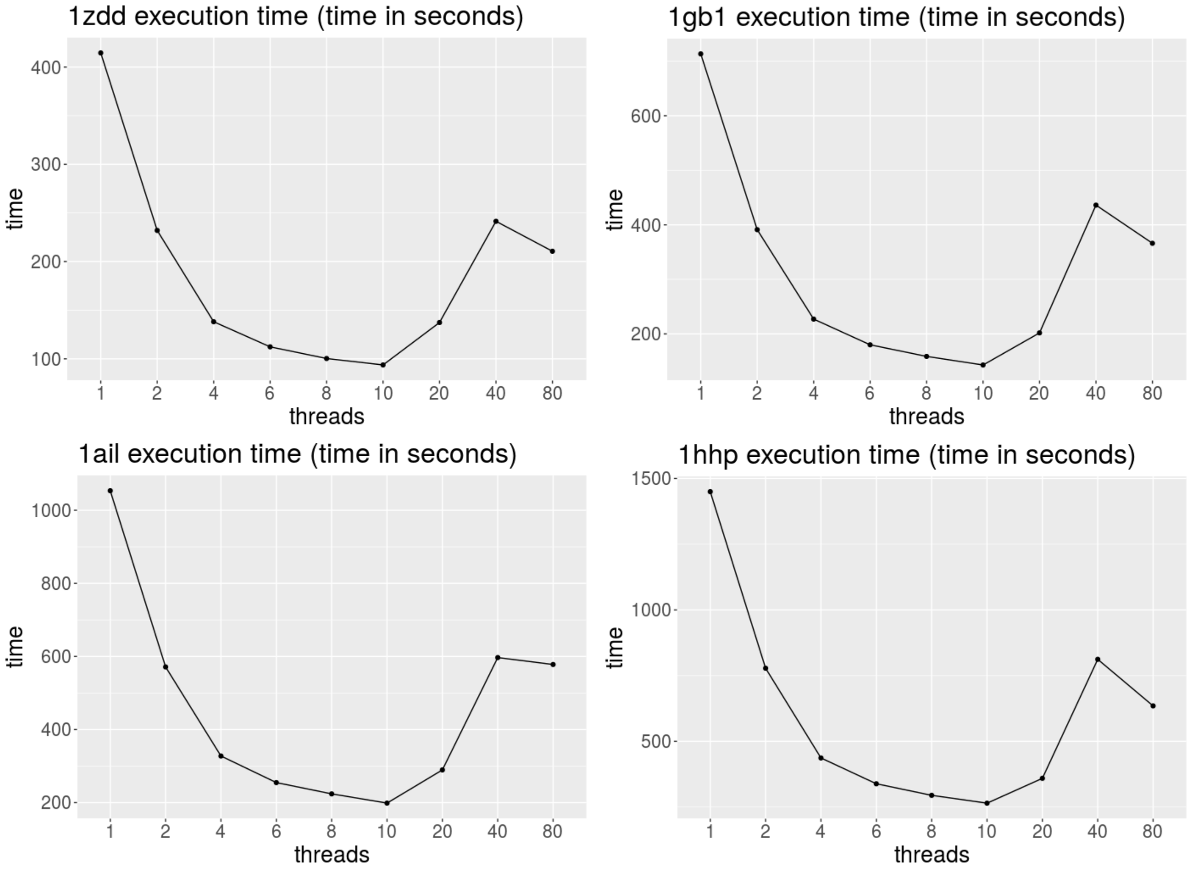
Source: Author

Table 7 – Average time in seconds for the controlled parallel execution

| | **1ZDD** | **1GB1** | **1AIL** | **1HHP** |
|---|---|---|---|---|
| **1** | 420.1 ± 0.9 | 728.1 ± 3.2 | 1060.2 ± 17.3 | 1468.6 ± 20.0 |
| **2** | 234.2 ± 1.4 | 395.0 ± 2.9 | 578.4 ± 9.8 | 780.1 ± 14.5 |
| **4** | 140.2 ± 0.6 | 229.2 ± 1.6 | 330.5 ± 3.3 | 438.2 ± 11.1 |
| **6** | 112.3 ± 0.7 | 179.9 ± 1.1 | 270.2 ± 3.4 | 350.9 ± 5.8 |
| **8** | 100.3 ± 0.4 | 158.6 ± 1.1 | 227.1 ± 2.9 | 297.4 ± 5.4 |
| **10** | 93.7 ± 0.4 | 201.8 ± 2.0 | 200.8 ± 1.4 | 269.5 ± 6.4 |
| **20** | 140.5 ± 0.8 | 145.4 ± 1.3 | 294.2 ± 3.1 | 363.2 ± 6.6 |
| **40** | 241.4 ± 70.4 | 436.1 ± 65.6 | 597.2 ± 103.4 | 815.0 ± 103.4 |
| **80** | 210.6 ± 44.7 | 366.1 ± 88.6 | 579.5 ± 138.5 | 637.9 ± 133.7 |

Source: Author

Figure 12 – Plot of the average execution time



Source: Author

### 5.2.3 Optimizer performance analysis

In this experiment, the performance of the proposed optimizer is analyzed. The search capability of the optimizer is measured using convergence and diversity metrics. This experiment used six proteins from Table 3 and are shown in Table 8. These proteins represent a small (lower than 100 amino acids) and large protein of each class.

Table 8 – Selected proteins for optimizer performance analysis

| Protein | Size | Class |
|---|---|---|
| 1DFN | 30 | $\beta$ |
| 1CRN | 46 | $\alpha\beta$ |
| 1AIL | 73 | $\alpha$ |
| T0968s1 | 119 | $\alpha\beta$ |
| T0868 | 161 | $\alpha$ |
| T1010 | 210 | $\beta$ |

Source: Author

To visualize the optimized Pareto fronts, scatter plots were utilized. For the convergence and diversity metrics, a line plot was utilized using the mean of 20 executions. The convergence metric is the hypervolume, described on Section 2.5 (Equation 6), and the diversity is the pairwise metric described on Section 2.5 (Equation 7). These metrics were calculated for the population of each generation.

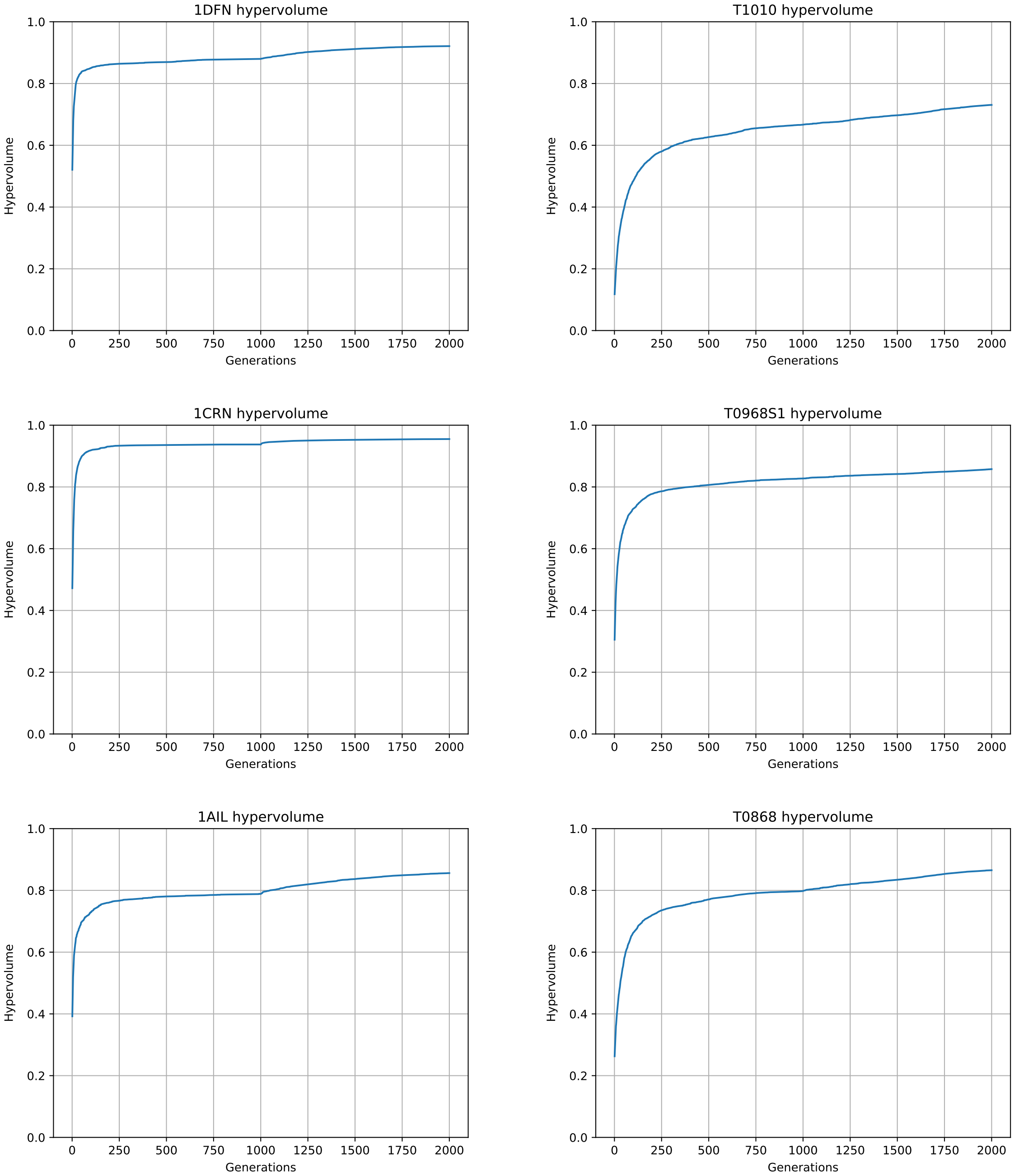
Figure 13 shows the convergence for all proteins. Overall, the main factor that impacts the convergence of the algorithm seems to be the protein size. All proteins exhibit similar behavior in the convergence of the MO-BRKGA/FRAG phase, converging to some frontier structure using few iterations. The MO-BRKGA/RES is then responsible for further evolving this frontier found by the MO-BRKGA/FRAG.

The premature convergence of the MO-BRKGA/FRAG is due to the use of fragments to build structures. As these fragments should be close to the native structure, the search space in this phase is relatively small. This also indicates a lower Pareto frontier diversity. That is, the objectives will not change much by changing fragments of the structure. Due to these considerations, a smaller number of iterations should be enough for this phase, leaving most of the optimization for the MO-BRKGA/RES.

Figure 14 shows the diversity for all proteins. It is possible to see that all proteins exhibit the same behavior. As defined by the parameter control of the algorithm, each optimization has two stages: exploration and exploitation. Overall, all proteins start at the exploration threshold of 0.4, and at half iterations, the optimization starts to converge. This convergence depends on the protein, falling to some diversity value and staying until the end. This behavior happens for both MO-BRKGA/FRAG and MO-BRKGA/RES, with the latter starting at the exploration threshold of 0.25.

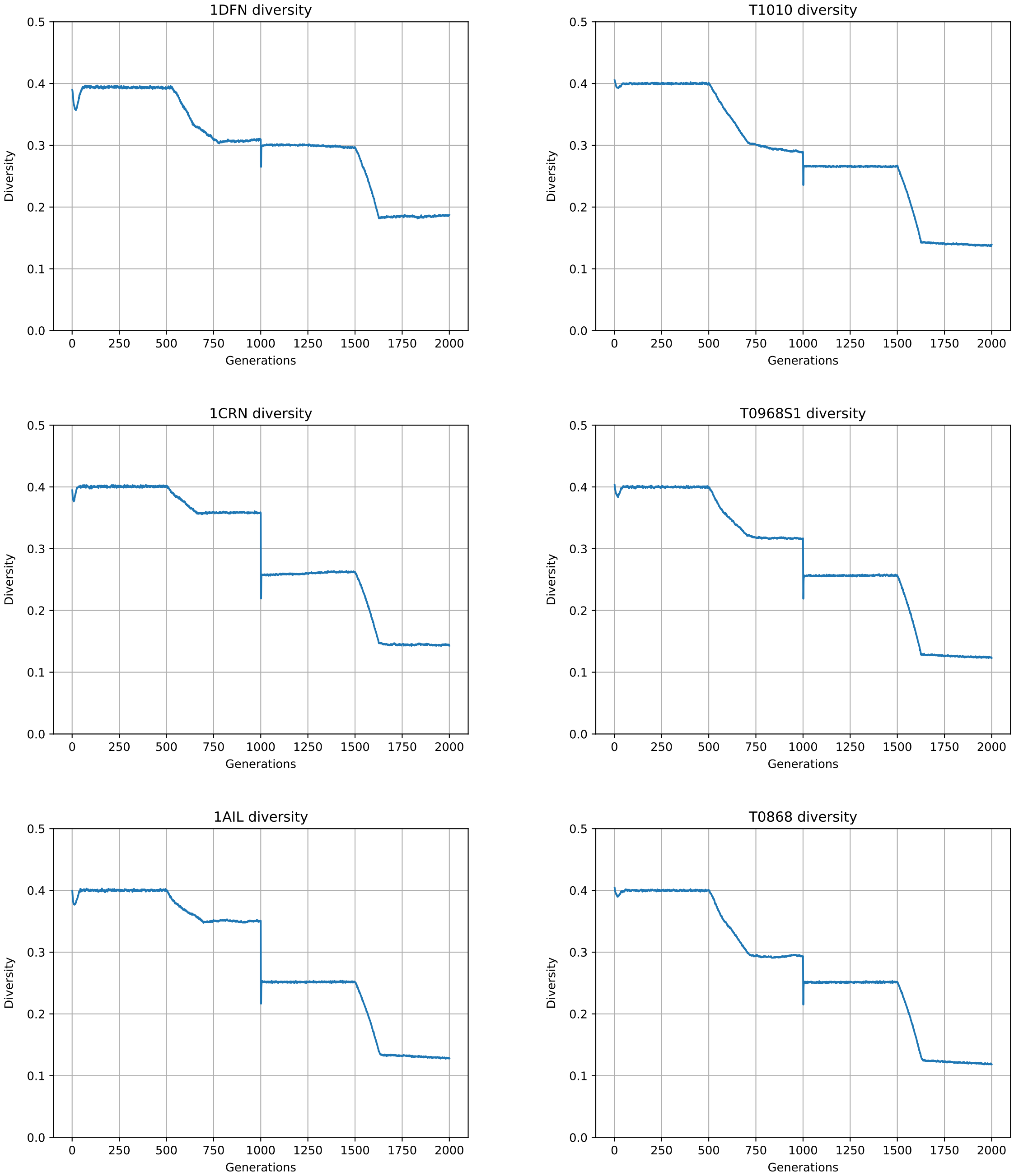
The following sections will explore the specific results of each protein. For each protein, scatter plots representing snapshots of the initial and final generation of both optimizer phases (MO-BRKGA/FRAG and RES) were generated. These plots display the three objectives in two dimensions: the X-axis showing the normalized secondary structure objective and the Y-axis showing the normalized contact map objective. The normalized energy score objective is represented by colors, using a color scale to define values.

Figure 13 – Proteins convergence



Source: Author

Figure 14 – Proteins diversity



Source: Author

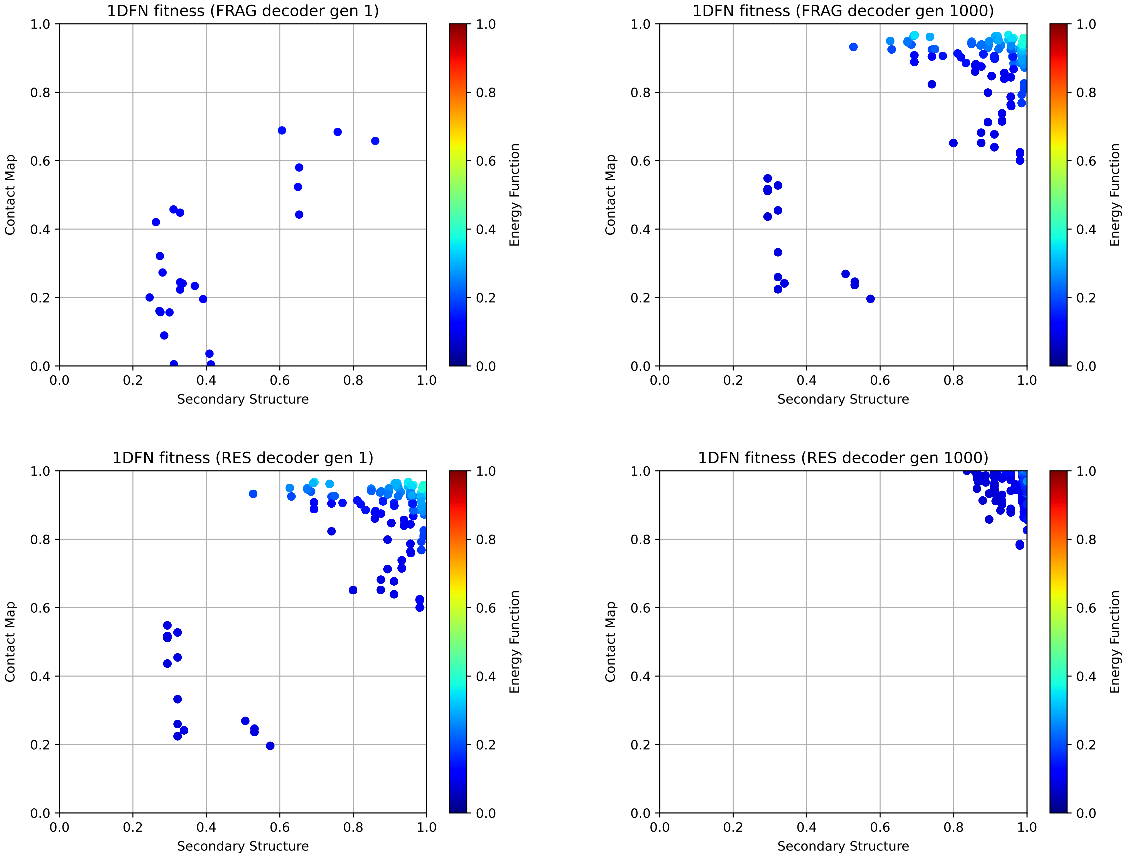
### *5.2.3.1 1DFN*

The protein 1DFN has an amino acid sequence of length 30 and is part of the $\beta$-class of proteins, which means it is mostly formed of $\beta$-sheets. The scatter plots of the optimization can be seen in Figure 15.

For this protein, the algorithm starts with a dispersed frontier (MO-BRKGA/FRAG generation 1) and converges to a frontier with almost optimal value for all objectives (MO-BRKGA/FRAG generation 1000). The MO-BRKGA/RES starts with the final frontier of the

MO-BRKGA/FRAG and is able to further optimize the frontier, reaching an almost optimal solution considering all objectives.

Figure 15 – 1DFN scatter plot



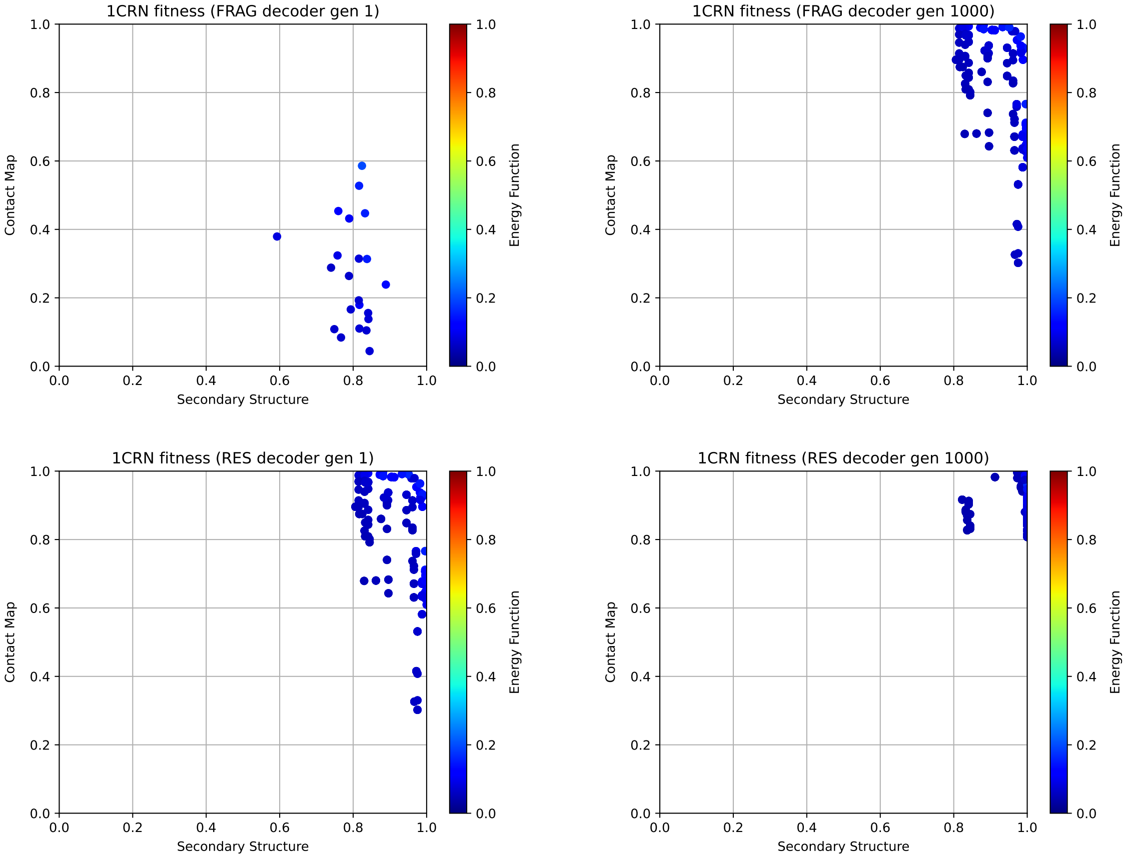
Source: Author

### 5.2.3.2 *1CRN*

The protein 1CRN has an amino acid sequence of length 46 and is part of the $\alpha\beta$-class of proteins, which means it is formed of $\alpha$-helices and $\beta$-sheets. The scatter plots of the optimization can be seen in Figure 16.

For this protein, the algorithm starts with a frontier close to the secondary structure optima (MO-BRKGA/FRAG generation 1) and converges to a frontier close to the optimal value of all objectives (MO-BRKGA/FRAG generation 1000). The MO-BRKGA/RES starts with the final frontier of the MO-BRKGA/FRAG and is able to further optimize all objectives, reaching an almost optimal solution considering all objectives.

Figure 16 – 1CRN scatter plot



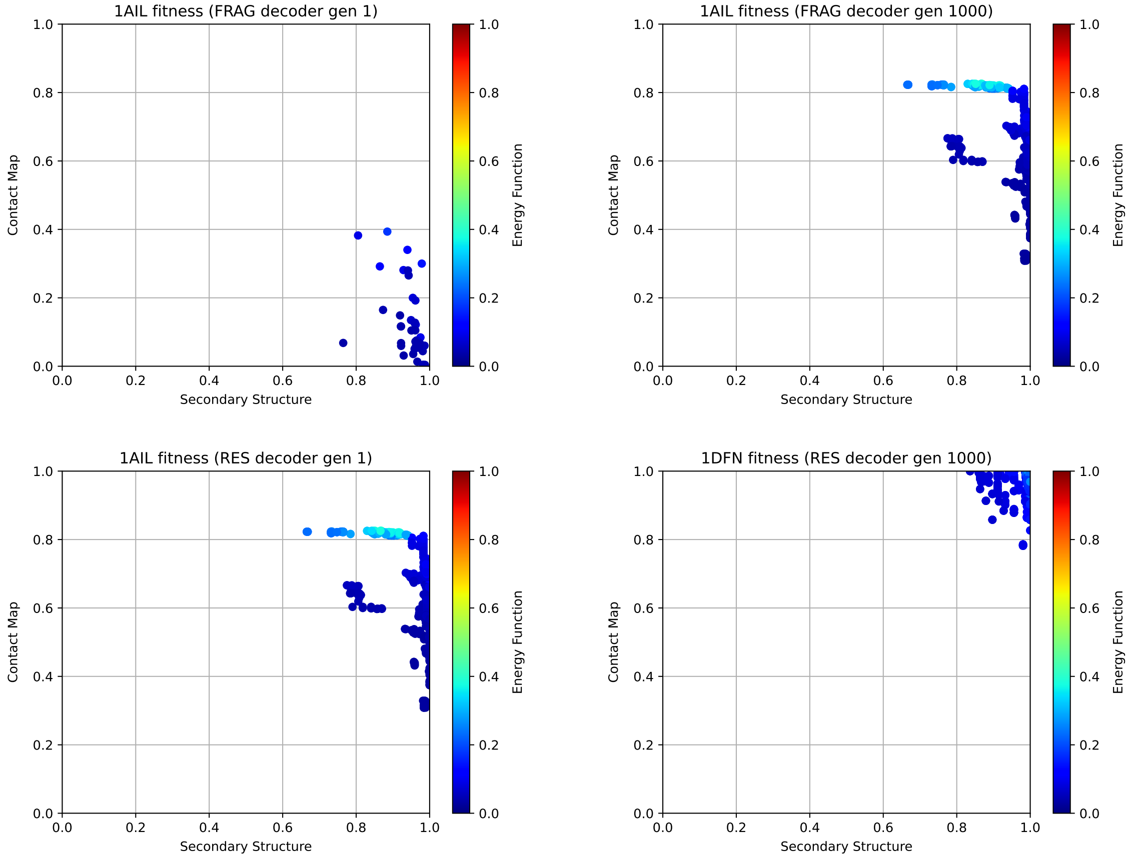
Source: Author

#### *5.2.3.3 1AIL*

The protein 1AIL has an amino acid sequence of length 73 and is part of the $\alpha$-class of proteins, which means it is formed mostly of $\alpha$-helices. The scatter plots of the optimization can be seen in Figure 17.

For this protein, the algorithm starts with a frontier near the secondary structure optima (MO-BRKGA/FRAG generation 1) and converges to a frontier close to the optimal value of all objectives (MO-BRKGA/FRAG generation 1000). The MO-BRKGA/RES starts with the final frontier of the MO-BRKGA/FRAG and is able to further optimize all objectives, reaching an almost optimal solution considering all objectives.

Figure 17 – 1AIL scatter plot



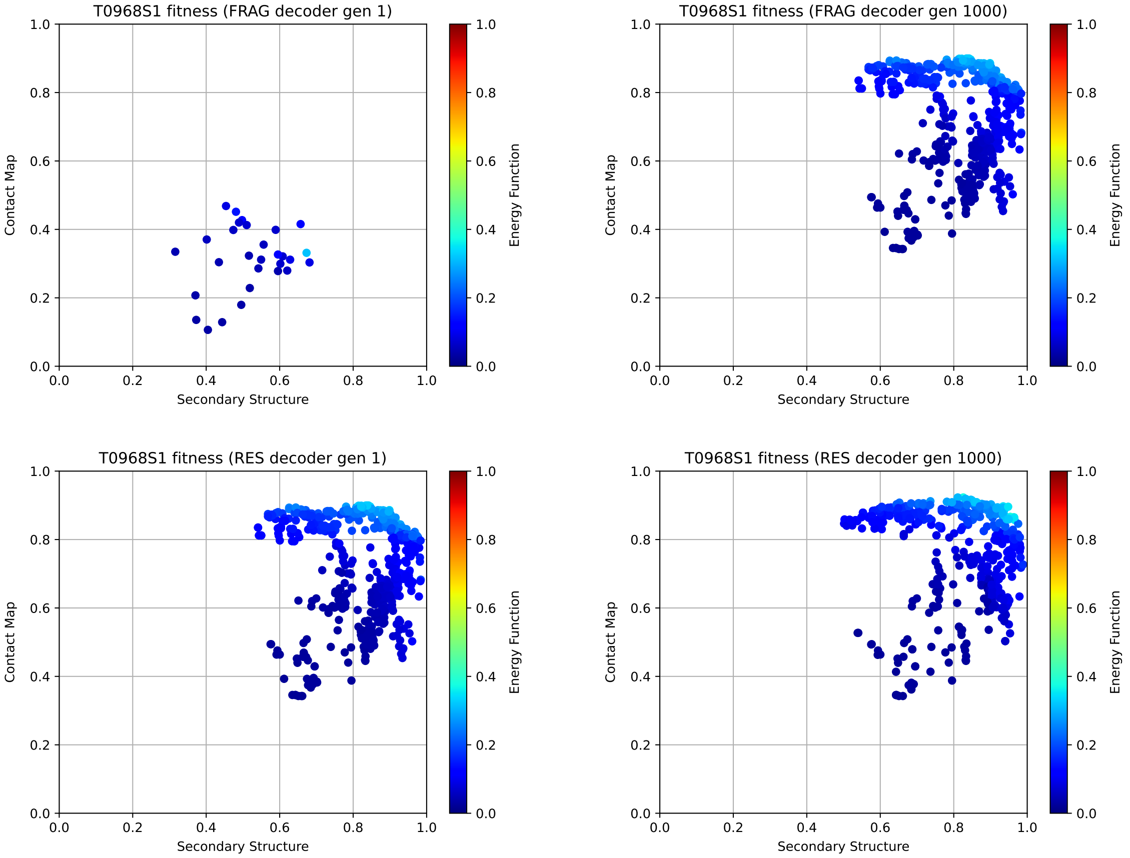
Source: Author

### *5.2.3.4 T0968s1*

The protein T0968s1 has an amino acid sequence of length 119 and is part of the $\alpha\beta$-class of proteins, which means it is formed of $\alpha$-helices and $\beta$-sheets. The scatter plots of the optimization can be seen in Figure 18.

For the large proteins (T0968s1, T0868, T1010), it is possible to observe more complex frontiers than the previous smaller proteins. Considering this protein, the algorithm starts with a frontier close to the secondary structure optima (MO-BRKGA/FRAG generation 1) and converges to a frontier with a clear trade-off between the objectives (MO-BRKGA/FRAG generation 1000). The MO-BRKGA/RES starts with the final frontier of the MO-BRKGA/FRAG and is able to slightly optimize the frontier.

Figure 18 – T0968s1 scatter plot



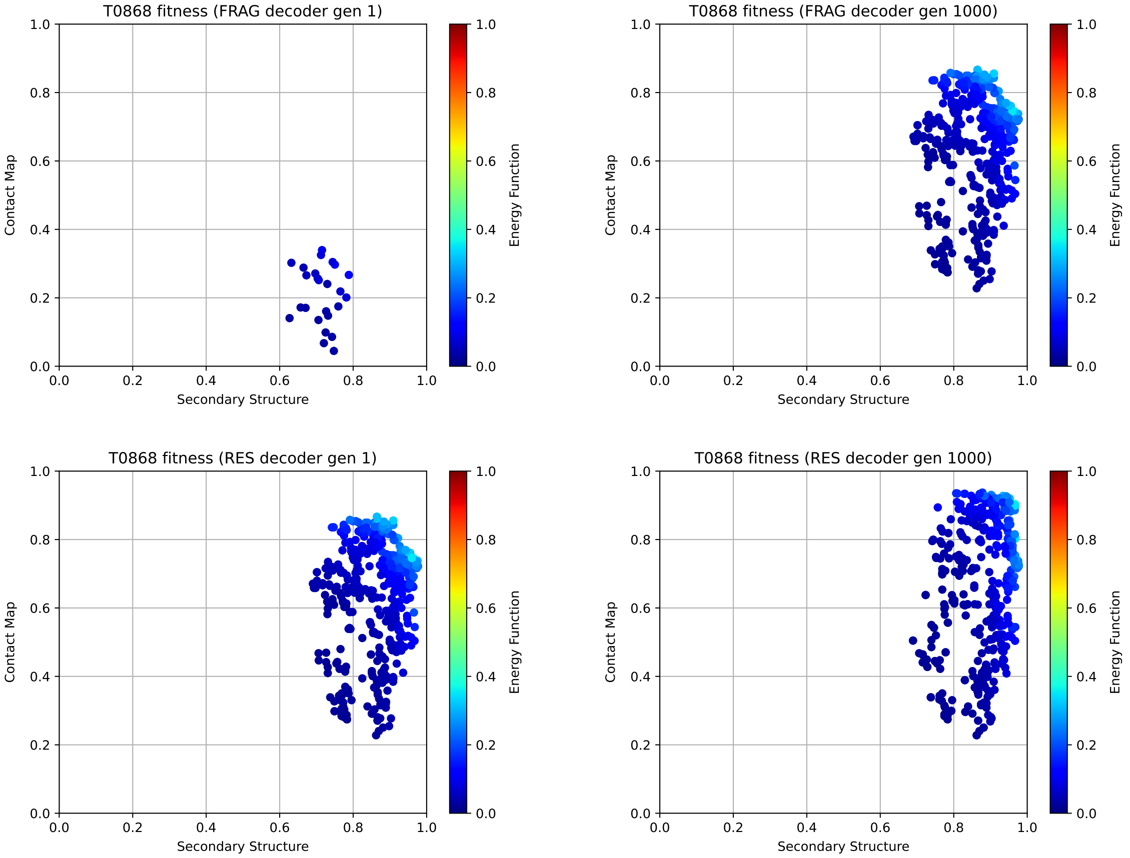
Source: Author

#### *5.2.3.5 T0868*

The protein T0868 has an amino acid sequence of length 161 and is part of the $\alpha$-class of proteins, which means it is formed mostly of $\alpha$-helices. The scatter plots of the optimization can be seen in Figure 19.

For this protein, the algorithm starts with a frontier close to the secondary structure optima (MO-BRKGA/FRAG generation 1) and converges (MO-BRKGA/FRAG generation 1000) to a complex frontier different from T0968s1. This distinct frontier structure could be due to the different protein classes. The MO-BRKGA/RES starts with the final frontier of the MO-BRKGA/FRAG and is able to further optimize the contact map objective, reaching solutions close to the optimal value for both contact map and secondary structure objectives.

Figure 19 – T0868 scatter plot



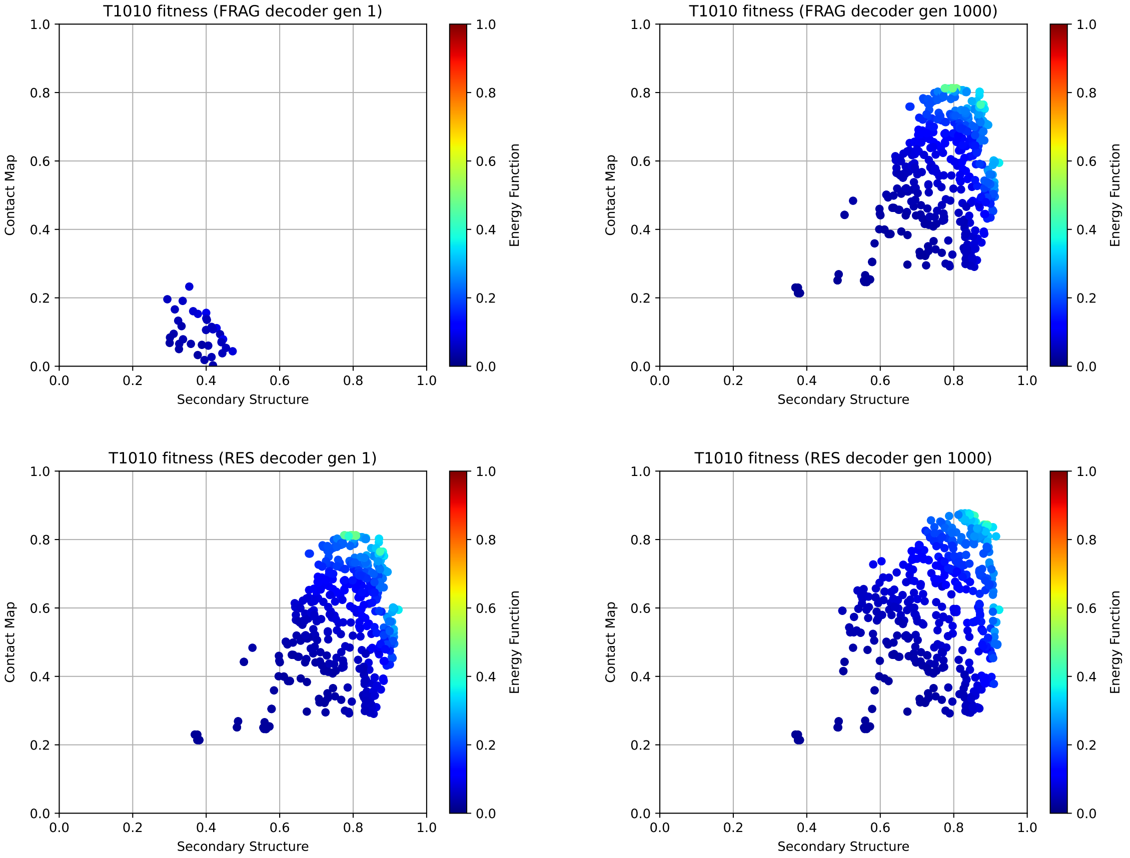
Source: Author

### 5.2.3.6 T1010

The protein T1010 has an amino acid sequence of length 210 and is part of the $\beta$-class of proteins, which means it is formed mostly of $\beta$-helices. The scatter plots of the optimization can be seen in Figure 20.

For this protein, the algorithm starts with a frontier with low energy but far from the secondary structure and contact map optima (MO-BRKGA/FRAG generation 1). The MO-BRKGA/FRAG converges (MO-BRKGA/FRAG generation 1000) to a complex frontier that has a different structure from T0968s1 and T0868. The MO-BRKGA/RES starts with the final frontier of the MO-BRKGA/FRAG and is able to slightly optimize the frontier.

Figure 20 – T1010 scatter plot



Source: Author

### 5.2.4 Predictor performance analysis

In this experiment, the performance of the predictor results is analyzed. This experiment uses all proteins listed in Table 3. The metrics introduced in Section 2.2.3 were utilized to measure the quality of the predicted structures. The best and average values (with standard deviation) of each metric were measured for all proteins.

Considering the proposed predictor, three types of results are analyzed:

- Best value of the final archive;
- Average value of the final archive; and
- Value from the structure selected with MUFOLD-CL.

This division was performed to analyze the overall quality of generated archive and the quality of the solution selected with MUFOLD-CL. Although it is not expected that the decision-maker will always select the best-generated structure, it should at least select a structure with quality higher than the average generated structure.

Comparisons with predictors from the literature are also conducted, which can be seen in Table 9. The results of the other predictors were extracted directly from their respective works.

The works of Chen et al. (2020), Narloch, Krause e Dorn (2020), and Song et al. (2018b) were presented in Section 3. The SCDE algorithm proposed by Zhang et al. (2018) is a single-objective optimizer for the PSP problem combining the information of secondary structures and contact maps. The same set of proteins was also predicted using the single-objective Rosetta *de novo* protocol (ROHL et al., 2004).

The overall results of both metrics were organized in Tables 10 and 11. The ANOVA test was performed to validate the statistical difference between the result of the proposed predictor MO-BRKGA (with MUFOLD-CL) and the other predictors. At the end of Tables 10 and 11, an extra row (B/S/W) was added to summarize the result of this test. B represents the number of proteins where the competing method was statistically better than the proposed predictor, and the W is the number of times where the competing method was statistically worse than the proposed predictor. S is the number of proteins where there was no statistical difference between the results of the competing method and the proposed predictor.

Table 9 – Compared methods

| **Reference** | **Algorithm** | **Function evaluations** |
|---|---|---|
| Chen et al. (2020) | MODE-K | 100,000 ($10^5$) |
| Narloch, Krause e Dorn (2020) | NSGA-II<br>GDE3<br>DEMO | 1,000,000 ($10^6$) |
| Song et al. (2018b) | MOPSO | 100,000 ($10^5$) |
| Zhang et al. (2018) | SCDE | 900,000 ($9 * 10^5$) |
| Rosetta | *de novo* protocol | 1,000,000 ($10^6$) |
| **Proposal** | **MO-BRKGA** | **1,000,000** ($10^6$) |

Source: Author

#### *5.2.4.1 RMSD*

The results for the RMSD metric are displayed in Table 10. The table compares each protein with the results of all the methods. The best and average values were utilized, if available. Missing results are marked with '-' in the table.

Considering the best solution found on all executions, the MO-BRKGA was able to generate the best solution for all proteins except 1ACW (Rosetta), 1ENH (SCDE), 1GB1 (Rosetta), I16C (SCDE), 1ZDD (DEMO), and 2P81 (MODE-K). The quality of the solutions found using MUFOLD-CL was not far from the best solution generated by the MO-BRKGA. It seems that, on average, the distance between the best solution and the MUFOLD-CL solution is between 1 and 3 Å.

Table 14 in Appendix A shows the results of the ANOVA test for the RMSD value of the MO-BRKGA with MUFOLD-CL. These results were used to create the B/S/W row in Table 10.

Overall, the solutions selected by the MUFOLD-CL are better than the other methods, except for SCDE. The MO-BRKGA with MUFOLD-CL selected better solutions most of the time when compared with NSGA2, GDE3, DEMO, and Rosetta. However, the method was better for only 1 out of 6 proteins when compared to SCDE.

Considering the best and mean values of the MO-BRKGA, the MUFOLD-CL was always worst than the best and almost always similar to the mean. These results make sense, as the MUFOLD-CL is a clustering method and selects as the final result the center of the largest cluster. As this center is similar to all the other structures of the cluster, and the largest cluster is selected, it is expected that the results are close to the mean of the final frontier.

Table 10 – RMSD results. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| Protein | | NSGA2 | GDE3 | DEMO | MOPSO | MODE-K | SCDE | Rosetta | MO-BRKGA Best | MO-BRKGA Mean | MO-BRKGA MUFOLD-CL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1AB1 | $f^*$ | - | - | - | 9.80 | 7.38 | - | 4.90 | **2.17** | 4.72 | 2.63 |
| | $\bar{x}$ | - | - | - | - | - | - | 7.78 | 2.93 | 5.81 | 5.08 |
| | $s$ | - | - | - | - | - | - | 1.20 | 0.46 | 0.75 | 1.61 |
| 1ACW | $f^*$ | 3.81 | 3.63 | 3.82 | - | - | - | **1.65** | 2.71 | 4.47 | 3.86 |
| | $\bar{x}$ | 6.47 | 6.56 | 7.17 | - | - | - | 5.48 | 3.90 | 5.55 | 5.29 |
| | $s$ | 1.58 | 1.72 | 1.81 | - | - | - | 1.27 | 0.58 | 0.44 | 0.55 |
| 1AIL | $f^*$ | 7.07 | 3.25 | 3.14 | - | - | **2.67** | 6.26 | 3.84 | 5.96 | 5.10 |
| | $\bar{x}$ | 10.30 | 6.77 | 7.40 | - | - | 3.00 | 9.50 | 5.72 | 7.78 | 7.42 |
| | $s$ | 1.38 | 2.81 | 2.55 | - | - | 0.17 | 1.98 | 0.89 | 0.81 | 1.31 |
| 1ALY | $f^*$ | - | - | - | - | - | 11.53 | 13.62 | **7.78** | 14.29 | 8.52 |
| | $\bar{x}$ | - | - | - | - | - | 11.77 | 16.51 | 11.29 | 17.29 | 14.59 |
| | $s$ | - | - | - | - | - | 0.49 | 2.14 | 1.75 | 1.29 | 2.93 |
| 1BDD | $f^*$ | - | - | - | 5.64 | 4.98 | - | 5.27 | **2.79** | 3.76 | 3.24 |
| | $\bar{x}$ | - | - | - | - | - | - | 7.61 | 3.26 | 4.15 | 3.94 |
| | $s$ | - | - | - | - | - | - | 1.60 | 0.19 | 0.29 | 0.59 |
| 1CRN | $f^*$ | 6.23 | 5.13 | 6.32 | 7.57 | - | - | 4.91 | **1.65** | 3.53 | 1.86 |
| | $\bar{x}$ | 9.24 | 9.62 | 9.31 | - | - | - | 7.24 | 2.52 | 5.28 | 4.72 |
| | $s$ | 1.50 | 2.69 | 2.79 | - | - | - | 1.02 | 0.45 | 1.04 | 2.08 |
| 1DFN | $f^*$ | - | - | - | - | 7.00 | - | 5.48 | **2.68** | 4.49 | 3.55 |
| | $\bar{x}$ | - | - | - | - | - | - | 7.10 | 3.23 | 5.12 | 4.72 |
| | $s$ | - | - | - | - | - | - | 0.68 | 0.34 | 0.57 | 1.05 |

Table 10 – RMSD results. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| Protein | | NSGA2 | GDE3 | DEMO | MOPSO | MODE-K | SCDE | Rosetta | MO-BRKGA Best | MO-BRKGA Mean | MO-BRKGA MUFOLD-CL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1ENH | $f^*$ | 6.84 | 3.10 | 4.29 | 8.92 | 7.80 | **1.12** | 3.66 | 1.94 | 3.29 | 2.93 |
| | $\bar{x}$ | 10.14 | 8.09 | 7.47 | - | - | 1.29 | 5.00 | 2.30 | 3.86 | 3.67 |
| | $s$ | 1.31 | 2.79 | 1.67 | - | - | 0.18 | 1.03 | 0.26 | 0.33 | 0.62 |
| 1GB1 | $f^*$ | - | - | - | - | - | 2.70 | **1.63** | **1.63** | 2.53 | 1.94 |
| | $\bar{x}$ | - | - | - | - | - | 3.40 | 2.92 | 2.45 | 3.38 | 2.94 |
| | $s$ | - | - | - | - | - | 0.41 | 1.82 | 0.53 | 0.63 | 0.74 |
| 1HHP | $f^*$ | - | - | - | - | - | 7.48 | 11.32 | **4.28** | 8.83 | 6.21 |
| | $\bar{x}$ | - | - | - | - | - | 8.96 | 14.58 | 8.57 | 12.80 | 10.55 |
| | $s$ | - | - | - | - | - | 0.34 | 1.28 | 2.26 | 1.93 | 2.63 |
| 1I6C | $f^*$ | - | - | - | 8.47 | 7.76 | **2.69** | 6.84 | 3.21 | 4.66 | 3.77 |
| | $\bar{x}$ | - | - | - | - | - | 3.69 | 8.31 | 3.74 | 5.37 | 5.32 |
| | $s$ | - | - | - | - | - | 0.34 | 0.76 | 0.41 | 0.35 | 0.76 |
| 1ROP | $f^*$ | 5.96 | 2.74 | 3.04 | 3.51 | 3.01 | - | 8.34 | **1.14** | 2.56 | 2.54 |
| | $\bar{x}$ | 10.86 | 7.05 | 6.80 | - | - | - | 11.01 | 1.63 | 3.56 | 3.18 |
| | $s$ | 1.85 | 2.07 | 1.61 | - | - | - | 1.52 | 0.43 | 0.99 | 0.52 |
| 1ZDD | $f^*$ | 3.47 | 1.79 | **1.19** | 2.15 | 2.50 | - | 2.99 | 1.31 | 1.54 | 1.55 |
| | $\bar{x}$ | 6.04 | 4.05 | 4.01 | - | - | - | 5.26 | 1.58 | 1.83 | 1.83 |
| | $s$ | 1.29 | 1.16 | 1.98 | - | - | - | 1.06 | 0.18 | 0.22 | 0.21 |
| 2KDL | $f^*$ | - | - | - | 10.29 | 7.72 | - | 11.41 | **6.01** | 10.57 | 10.07 |
| | $\bar{x}$ | - | - | - | - | - | - | 12.98 | 8.55 | 11.26 | 11.13 |
| | $s$ | - | - | - | - | - | - | 0.44 | 0.97 | 0.37 | 0.67 |

Table 10 – RMSD results. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| **Protein** | | **NSGA2** | **GDE3** | **DEMO** | **MOPSO** | **MODE-K** | **SCDE** | **Rosetta** | **MO-BRKGA Best** | **MO-BRKGA Mean** | **MO-BRKGA MUFOLD-CL** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2MR9 | $f^*$ | 6.16 | 3.00 | 2.62 | - | - | - | 2.21 | **1.66** | 2.41 | 2.37 |
| | $\bar{x}$ | 8.29 | 7.09 | 7.21 | - | - | - | 4.46 | 1.89 | 2.61 | 2.61 |
| | $s$ | 1.29 | 1.87 | 1.87 | - | - | - | 2.62 | 0.20 | 0.12 | 0.18 |
| 2P81 | $f^*$ | 5.85 | 4.78 | 5.06 | 6.28 | **4.76** | - | 5.56 | 5.89 | 8.33 | 7.94 |
| | $\bar{x}$ | 8.94 | 7.10 | 7.72 | - | - | - | 7.97 | 7.12 | 9.08 | 8.94 |
| | $s$ | 1.30 | 1.34 | 1.44 | - | - | - | 1.34 | 0.67 | 0.37 | 0.49 |
| T0868 | $f^*$ | - | - | - | - | - | - | 13.06 | **3.38** | 6.00 | 4.16 |
| | $\bar{x}$ | - | - | - | - | - | - | 15.26 | 5.20 | 8.82 | 6.90 |
| | $s$ | - | - | - | - | - | - | 1.41 | 1.15 | 1.46 | 1.63 |
| T0900 | $f^*$ | - | - | - | - | - | - | 12.71 | **5.93** | 10.72 | 7.43 |
| | $\bar{x}$ | - | - | - | - | - | - | 15.07 | 7.39 | 12.21 | 9.40 |
| | $s$ | - | - | - | - | - | - | 0.88 | 0.74 | 0.66 | 1.33 |
| T0968S1 | $f^*$ | - | - | - | - | - | - | 12.79 | **4.85** | 8.47 | 5.65 |
| | $\bar{x}$ | - | - | - | - | - | - | 15.78 | 6.50 | 11.43 | 8.93 |
| | $s$ | - | - | - | - | - | - | 1.67 | 1.02 | 1.49 | 2.00 |
| T1010 | $f^*$ | - | - | - | - | - | - | 18.37 | **12.86** | 20.68 | 14.90 |
| | $\bar{x}$ | - | - | - | - | - | - | 21.13 | 14.36 | 22.50 | 19.46 |
| | $s$ | - | - | - | - | - | - | 1.43 | 0.88 | 0.97 | 3.53 |
| **B/S/W** | | 0/1/7 | 1/1/6 | 1/1/6 | - | - | 5/0/1 | 1/3/16 | 20/0/0 | 0/14/6 | - |

Source: Author

#### *5.2.4.2 GDT_TS*

The results for the GDT_TS metric are displayed in Table 11 in Appendix A. The table compares each protein with the results of all the methods. The best and average values were utilized, if available. Missing results are marked with **-** in the table.

Considering the best solution found on all executions, the MO-BRKGA was able to generate the best solution for all proteins except 1ACW (Rosetta) and 1GB1 (Rosetta). On average, the distance between the best solution and the MUFOLD-CL solution seems to be between 10 and 20.

Table 15 shows the results of the ANOVA test for the GDT_TS value of the MO-BRKGA with MUFOLD-CL. These results were used to create the B/S/W row in Table 11.

The results are similar to the results of the RMSD metric. Overall, the solutions selected by the MUFOLD-CL are better than the other methods. The MO-BRKGA with MUFOLD-CL selected better solutions most of the time when compared with NSGA2, GDE3, DEMO, and Rosetta. In this case, the SCDE method was not considered as it did not have GDT_TS values. Again, the solutions selected with MUFOLD-CL are worse than the best and close to the mean value of the final frontier generated by the MO-BRKGA.

Table 11 – GDT_TS results. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| **Protein** | | **NSGA2** | **GDE3** | **DEMO** | **MOPSO** | **MODE-K** | **SCDE** | **Rosetta** | **MO-BRKGA Best** | **MO-BRKGA Mean** | **MO-BRKGA MUFOLD-CL** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1AB1 | $f^*$ | - | - | - | 36.96 | 38.04 | - | 48.70 | **81.74** | 65.37 | 78.26 |
| | $\bar{x}$ | - | - | - | - | - | - | 34.98 | 76.91 | 60.06 | 65.28 |
| | $s$ | - | - | - | - | - | - | 6.05 | 2.90 | 5.35 | 8.97 |
| 1ACW | $f^*$ | 57.93 | 65.51 | 62.75 | - | - | - | **86.21** | 68.97 | 54.25 | 58.62 |
| | $\bar{x}$ | 43.70 | 50.75 | 48.89 | - | - | - | 49.14 | 57.31 | 44.56 | 46.07 |
| | $s$ | 7.94 | 6.03 | 6.07 | - | - | - | 11.12 | 6.34 | 3.64 | 5.32 |
| 1AIL | $f^*$ | 31.42 | **67.14** | 61.71 | - | - | - | 40.00 | 60.57 | 38.38 | 41.71 |
| | $\bar{x}$ | 22.91 | 48.10 | 48.80 | - | - | - | 30.49 | 44.34 | 34.01 | 34.26 |
| | $s$ | 3.54 | 8.36 | 6.97 | - | - | - | 4.35 | 5.36 | 2.24 | 3.37 |
| 1ALY | $f^*$ | - | - | - | - | - | - | 16.71 | **28.22** | 18.86 | 23.70 |
| | $\bar{x}$ | - | - | - | - | - | - | 12.18 | 21.43 | 14.77 | 16.23 |
| | $s$ | - | - | - | - | - | - | 1.88 | 3.37 | 1.87 | 3.28 |
| 1BDD | $f^*$ | - | - | - | 50.42 | 52.08 | - | 63.67 | **74.67** | 69.87 | 71.33 |
| | $\bar{x}$ | - | - | - | - | - | - | 47.68 | 70.55 | 65.23 | 66.15 |
| | $s$ | - | - | - | - | - | - | 7.55 | 2.65 | 2.04 | 2.69 |
| 1CRN | $f^*$ | 42.60 | 56.52 | 50.00 | 46.20 | - | - | 56.52 | **88.70** | 75.55 | 86.09 |
| | $\bar{x}$ | 31.04 | 40.55 | 38.81 | - | - | - | 43.09 | 82.43 | 66.81 | 71.46 |
| | $s$ | 4.58 | 7.08 | 4.71 | - | - | - | 6.70 | 2.70 | 4.46 | 10.46 |
| 1DFN | $f^*$ | - | - | - | - | 46.67 | - | 53.33 | **80.00** | 63.19 | 67.33 |
| | $\bar{x}$ | - | - | - | - | - | - | 44.60 | 70.33 | 56.50 | 57.13 |
| | $s$ | - | - | - | - | - | - | 6.85 | 5.32 | 3.80 | 5.72 |

Table 11 – GDT_TS results. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| Protein | | NSGA2 | GDE3 | DEMO | MOPSO | MODE-K | SCDE | Rosetta | MO-BRKGA Best | MO-BRKGA Mean | MO-BRKGA MUFOLD-CL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1ENH | $f^*$ | 39.62 | 73.70 | 72.96 | 46.30 | 42.13 | - | 78.89 | **85.19** | 72.81 | 78.52 |
| | $\bar{x}$ | 26.86 | 46.22 | 49.07 | - | - | - | 58.76 | 80.44 | 67.67 | 69.52 |
| | $s$ | 3.79 | 8.73 | 7.82 | - | - | - | 10.78 | 2.96 | 3.38 | 5.31 |
| 1GB1 | $f^*$ | - | - | - | - | - | - | 84.64 | **85.00** | 72.63 | 80.71 |
| | $\bar{x}$ | - | - | - | - | - | - | 71.91 | 76.64 | 66.24 | 69.59 |
| | $s$ | - | - | - | - | - | - | 11.29 | 4.13 | 3.86 | 5.79 |
| 1HHP | $f^*$ | - | - | - | - | - | - | 21.01 | **59.39** | 38.81 | 46.46 |
| | $\bar{x}$ | - | - | - | - | - | - | 16.24 | 39.30 | 27.33 | 31.19 |
| | $s$ | - | - | - | - | - | - | 2.48 | 7.96 | 4.76 | 7.76 |
| 1I6C | $f^*$ | - | - | - | 32.69 | 37.18 | - | 56.92 | **69.23** | 62.33 | 63.08 |
| | $\bar{x}$ | - | - | - | - | - | - | 49.90 | 64.79 | 57.54 | 58.56 |
| | $s$ | - | - | - | - | - | - | 5.42 | 2.55 | 1.88 | 2.61 |
| 1ROP | $f^*$ | 42.14 | 68.92 | 66.42 | 66.96 | 66.07 | - | 42.50 | **92.50** | 72.37 | 78.57 |
| | $\bar{x}$ | 26.65 | 45.22 | 46.01 | - | - | - | 36.34 | 84.48 | 64.12 | 64.04 |
| | $s$ | 5.33 | 8.21 | 6.96 | - | - | - | 4.07 | 5.31 | 5.87 | 7.44 |
| 1ZDD | $f^*$ | 61.17 | 87.05 | **93.52** | 77.94 | 77.21 | - | 77.06 | 91.76 | 87.32 | 87.65 |
| | $\bar{x}$ | 41.50 | 60.76 | 62.72 | - | - | - | 52.44 | 88.47 | 85.04 | 84.91 |
| | $s$ | 9.27 | 9.58 | 12.90 | - | - | - | 9.83 | 1.53 | 1.62 | 2.21 |
| 2KDL | $f^*$ | - | - | - | 41.52 | 42.41 | - | 37.86 | **46.43** | 38.53 | 41.07 |
| | $\bar{x}$ | - | - | - | - | - | - | 30.48 | 42.36 | 34.79 | 36.20 |
| | $s$ | - | - | - | - | - | - | 2.74 | 1.86 | 1.99 | 2.64 |

Table 11 – GDT_TS results. For each method, the best solution found ($f*$), mean value ($\bar{x}$) and standard deviation ($s$) are displayed. The best absolute $f*$ for each protein is in bold.

| Protein | | NSGA2 | GDE3 | DEMO | MOPSO | MODE-K | SCDE | Rosetta | MO-BRKGA Best | MO-BRKGA Mean | MO-BRKGA MUFOLD-CL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2MR9 | $f^*$ | 43.18 | 71.36 | 70.45 | - | - | - | 85.45 | **91.82** | 78.41 | 82.73 |
| | $\bar{x}$ | 32.34 | 48.10 | 48.57 | - | - | - | 66.86 | 87.95 | 75.44 | 75.66 |
| | $s$ | 4.94 | 8.41 | 8.82 | - | - | - | 15.16 | 2.33 | 1.88 | 3.14 |
| 2P81 | $f^*$ | 37.72 | **69.09** | 69.09 | 47.73 | 65.34 | - | 62.73 | 68.64 | 53.84 | 58.18 |
| | $\bar{x}$ | 30.50 | 53.51 | 49.81 | - | - | - | 52.11 | 62.98 | 49.47 | 53.00 |
| | $s$ | 4.12 | 6.06 | 6.61 | - | - | - | 6.51 | 3.38 | 3.17 | 3.73 |
| T0868 | $f^*$ | - | - | - | - | - | - | 15.00 | **60.17** | 47.49 | 51.38 |
| | $\bar{x}$ | - | - | - | - | - | - | 12.77 | 47.02 | 35.07 | 38.87 |
| | $s$ | - | - | - | - | - | - | 0.96 | 6.96 | 5.30 | 6.52 |
| T0900 | $f^*$ | - | - | - | - | - | - | 13.96 | **39.41** | 29.86 | 34.90 |
| | $\bar{x}$ | - | - | - | - | - | - | 11.95 | 30.08 | 21.38 | 22.86 |
| | $s$ | - | - | - | - | - | - | 1.19 | 4.50 | 2.82 | 4.09 |
| T0968S1 | $f^*$ | - | - | - | - | - | - | 17.29 | **51.02** | 34.66 | 42.88 |
| | $\bar{x}$ | - | - | - | - | - | - | 13.86 | 41.09 | 28.34 | 33.05 |
| | $s$ | - | - | - | - | - | - | 1.35 | 4.77 | 2.81 | 5.71 |
| T1010 | $f^*$ | - | - | - | - | - | - | 11.62 | **24.57** | 17.36 | 18.19 |
| | $\bar{x}$ | - | - | - | - | - | - | 9.05 | 18.41 | 12.92 | 14.08 |
| | $s$ | - | - | - | - | - | - | 1.52 | 2.84 | 1.69 | 2.48 |
| **B/S/W** | | 0/1/7 | 2/1/5 | 1/2/5 | - | - | - | 0/3/17 | 20/0/0 | 0/16/4 | - |

Source: Author

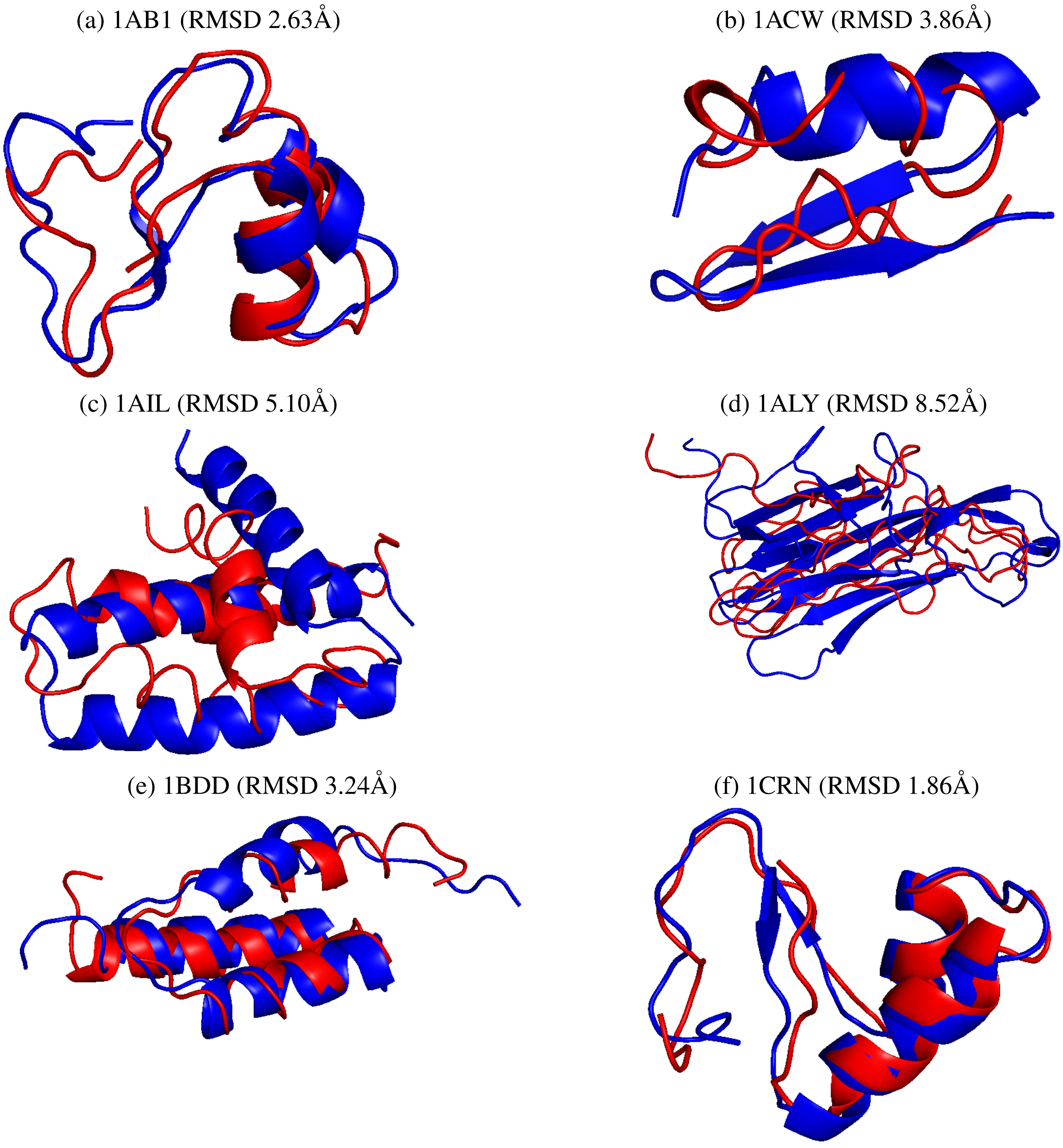
#### *5.2.4.3 Predicted structures visualization*

The visualization of the predicted structures can be seen in Figure 21. In this figure, the best solution predicted by the MO-BRKGA with MUFOLD-CL is in red, while the native structure is in blue. The structures were overlapped using the $\alpha$-carbons. This overlap displays the quality of the predicted structures.

It is possible to see that the $\alpha$-helices of the proteins were accurately predicted, in general. This behavior can be seen in the structures of 1AB1, 1BDD, 1CRN, 1ENH, 1GB1, 1ROP, 1ZDD, and 2MR9. However, for some $\alpha$ proteins, the algorithm was unable to generate accurate helices. This can be observed in 2KDL and 2P81, which are small $\alpha$ proteins with poor predictions. This divergence of quality is probably due to the secondary structure and contact map information predicted for these proteins. If the input information has poor quality, it is expected that the output structure will be equally bad.

It is also possible to see in this visualization that one difficult part is the prediction of $\beta$-sheets. These structures are harder to predict than the $\alpha$-helices, and proteins of the class $\beta$ are usually the ones with the lowest prediction quality. This difficulty can be observed both in simpler proteins, such as 1DFN, and in the more complex, such as 1ALY and 1HHP.

Figure 21 – Overlap of predicted and native figures. The predicted structure is in red and the native structure is in blue.



(a) 1AB1 (RMSD 2.63Å)

(b) 1ACW (RMSD 3.86Å)

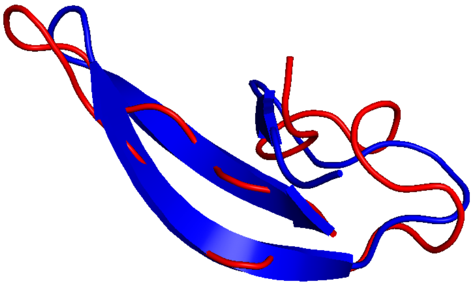
(c) 1AIL (RMSD 5.10Å)

(d) 1ALY (RMSD 8.52Å)

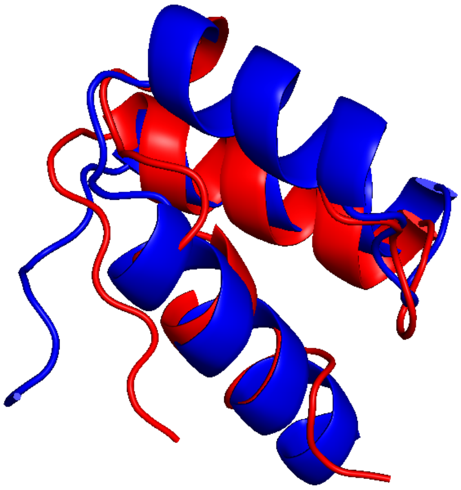
(e) 1BDD (RMSD 3.24Å)

(f) 1CRN (RMSD 1.86Å)

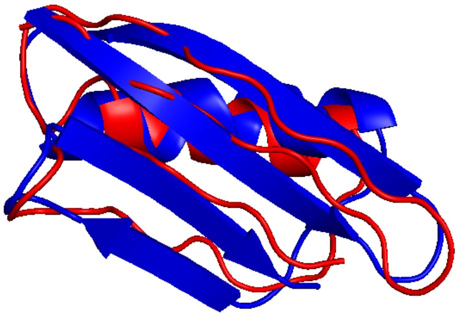
(g) 1DFN (RMSD 3.55Å)



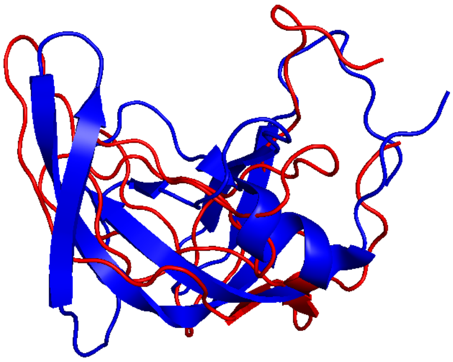
(h) 1ENH (RMSD 2.93Å)



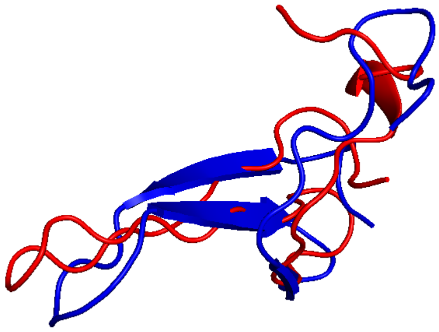
(i) 1GB1 (RMSD 1.94Å)



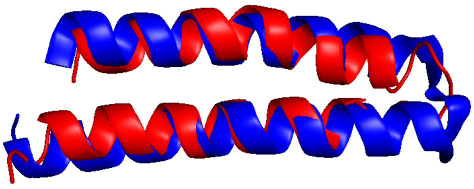
(j) 1HHP (RMSD 6.21Å)

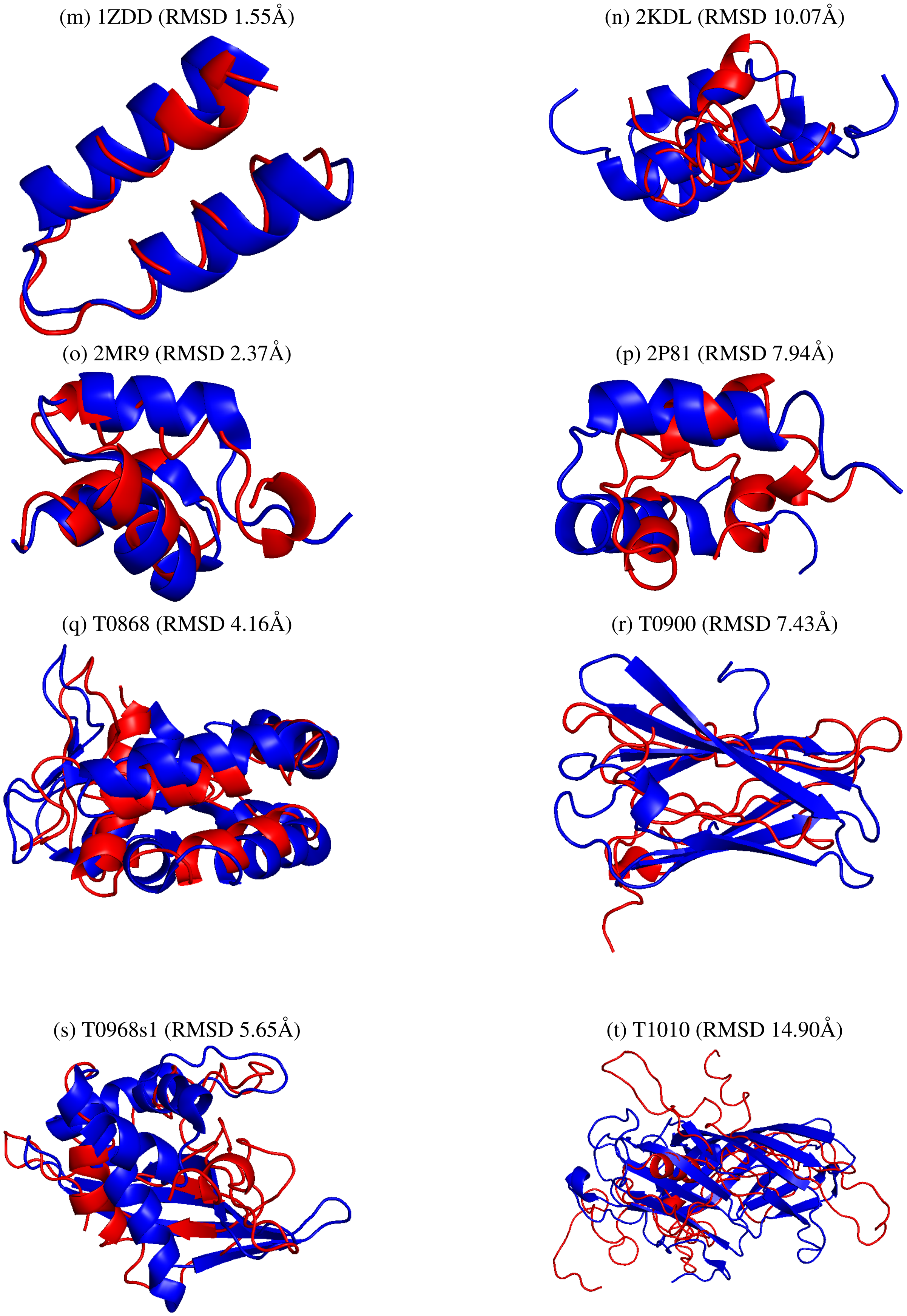


(k) 1I6C (RMSD 3.77Å)



(l) 1ROP (RMSD 2.54Å)





(m) 1ZDD (RMSD 1.55Å)

(n) 2KDL (RMSD 10.07Å)

(o) 2MR9 (RMSD 2.37Å)

(p) 2P81 (RMSD 7.94Å)

(q) T0868 (RMSD 4.16Å)

(r) T0900 (RMSD 7.43Å)

(s) T0968s1 (RMSD 5.65Å)

(t) T1010 (RMSD 14.90Å)

Source: Author

#### *5.2.4.4 Overall Analysis*

With the results of this set of experiments, it is possible to see the suitability of the proposed method. Overall, the results indicate that the proposed method is able to generate structures with competitive quality compared with the literature. Although the structures selected with the decision-maker are not the best, their quality is at least similar to the structures generated in the literature, which in most cases do not consider a decision-maker.

Considering the structures generated, the bottleneck of the algorithm is the prediction of $\beta$-sheets. Proteins of the class $\alpha$ had reasonable results, even for larger proteins such as T0868. In contrast, the prediction of $\beta$-sheets was hard, even for very small proteins such as 1DFN. Still, the proposed optimizer was able to generate reasonable structures for these proteins when compared with the selected works.

# 6 CONCLUSIONS AND FUTURE WORKS

Proteins are base molecules and very important in the functioning of live organisms. To better understand how proteins work, it is necessary to understand how their structures are formed. The process of formation of proteins structures is a highly complex biological process, which is still not completely known. Because of this, the protein structure prediction problem is one of the most important open problems of the bioinformatics area.

Computational methods can be used to solve this problem. The main paradigms are the homology, threading, and *ab initio* methods. Among these, the *ab initio* is the most interesting, as these methods do not require known similar structures to predict an unknown protein structure. The *ab initio* methods transform the prediction problem into an optimization, which usually has only a single optimization objective. It is known, however, that the PSP is better optimized as a multi-objective, given the conflicts existing between the multiple different information used to guide the optimization.

In this work, a multi-objective model and optimizer for the PSP problem were proposed. The optimization model consisted of three objectives: energy function, secondary structure information, and contact map information. The proposed optimizer was a BRKGA modified to solve multi-objective problems, named MO-BRKGA. This algorithm incorporated the use of non-dominated sorting, crowding distance, and archives. Also, an online parameter control technique was employed to reduce the number of parameters and utilize optimization information to increase the search efficiency of the algorithm.

Two decoders were utilized to balance the exploration and exploitation of the search space. The fragment decoder utilized fragments to generate structures, while the residue decoder generated structures in a continuous domain. The final predictor was composed of two phases: the MO-BRKGA with fragment decoder (MO-BRKGA/FRAG) and the MO-BRKGA with residue decoder (MO-BRKGA/RES). The results of the first phase were utilized as the initial population of the second phase, combining both decoders in a single search. To select a structure from the final archive (solution of the second phase), the clustering method MUFOLD-CL was used.

Different experiments were conducted to demonstrate the effectiveness of the proposed predictor. The first experiment, an execution time analysis, showed that the predictor is scalable with regard to CPU. The results demonstrated the effectiveness of properly using the available computer architecture, which in this work was a computer with NUMA architecture. By properly controlling the distribution of threads, a speedup of approximately 5 was achieved for 10 cores.

The second experiment was an analysis of the optimizer performance. It was possible to see that the proposed optimizer is capable of effectively explore the search space of proteins of different sizes and classes. However, it also indicated a small disturbance in the connection between the optimizer phases. An improvement of this connection should further increase the performance of the proposed algorithm.

The final experiment was the analysis of the predictor performance. The results of these experiments demonstrated that the complete predictor (with the decision-making step) is highly competitive with other works from the literature. However, the prediction of $\beta$-sheets is shown to be a weakness of the proposed predictor. Although the predictor is consistently able to predict $\alpha$-helices, the prediction of $\beta$-sheet is unstable, with proteins of the $\beta$ class having considerably lower quality when compared with proteins of the $\alpha$ class. A paper reporting the results obtained was presented on the IEEE Congress on Evolutionary Computation (IEEE CEC), 2021(MARCHI; PARPINELLI, 2021).

One difficulty observed is the inconsistency of results, where the size and class of the protein do not seem to be enough to accurately determine the effectiveness of the predictor. This happens because the quality of the results is directly associated with the quality of the input information.

The quality of fragments, secondary structure information, and contact map information is associated with their respective predictors. As these predictions are not exact, a considerable degree of uncertainty is present in the optimization. Poor predictions of this base information will result in poor results of the final predictor, regardless of the quality of the search process itself.

Considering the raised difficulties, it is possible to define some improvements in the proposed work. Multiple predictors could be used to reduce the inaccuracy of predicted information in the optimization. By using the information of different predictors, it is possible the reduce bias and combine the best information from each source. This combination of multiple data sources may increase the effectiveness of secondary structures and contact maps in the prediction of tertiary structures.

Also, the prediction of $\beta$ should be more thoroughly researched. Other types of information could be added to the optimization model with the objective of increasing the quality of $\beta$-sheets in protein predictions. By focusing on this weak point, the overall effectiveness of the predictor should increase not only for $\beta$ proteins but also for all proteins in general.

Another line of work is to further improve the search algorithm itself. This work presents a simple online parameter control that is able to select reasonable values. However, more complex techniques could be employed, such as the use of fuzzy systems to incorporate expert information about the problem that is being optimized. Also, local search could be implemented to further specialize the proposed algorithm for the optimization of protein structures.

It is also possible to extend the proposed model with other types of information. Although this work used the Rosetta centroid energy function, the literature demonstrates that most works use all-atoms energy. The computational complexity of these energy functions is considerably higher, although they are also more accurate. In particular, most of the observed works used the combination of bonded and non-bonded energies with other high-level information.

Finally, the exploration of these more complex models will also require modifications of the proposed algorithm. The proposed optimizer is able to work models of up to three objectives. Models with three or more objectives are denominated many-objective models and are considerably harder to optimize than multi-objective models. Considering the extension of the proposed model to incorporate more information, it is expected that the proposed algorithm is further modified to be able to optimize such many-objective models.

# BIBLIOGRAPHY

BRANKE, Jürgen et al. Finding knees in multi-objective optimization. In: SPRINGER. **International conference on parallel problem solving from nature**. Berlin, 2004. p. 722–731. Cited on page 30.

BRASIL, Christiane Regina Soares; DELBEM, Alexandre Claudio Botazzo; SILVA, Fernando Luís Barroso da. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. **Journal of computational chemistry**, Wiley Online Library, v. 34, n. 20, p. 1719–1734, 2013. Cited 2 times on pages 32 e 35.

CHEN, Xingqian et al. Incorporating a multiobjective knowledge-based energy function into differential evolution for protein structure prediction. **Information Sciences**, Elsevier, v. 540, p. 69–88, 2020. Cited 3 times on pages 34, 37 e 63.

CORRÊA, Leonardo de Lima; DORN, Márcio. A multi-objective swarm-based algorithm for the prediction of protein structures. In: SPRINGER. **International Conference on Computational Science**. Cham, 2019. p. 101–115. Cited 2 times on pages 33 e 37.

CUTELLO, Vincenzo; NARZISI, Giuseppe; NICOSIA, Giuseppe. A multi-objective evolutionary approach to the protein structure prediction problem. **Journal of The Royal Society Interface**, The Royal Society London, v. 3, n. 6, p. 139–151, 2006. Cited 5 times on pages 15, 30, 32, 35 e 38.

DEB, Kalyanmoy et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002. Cited 4 times on pages 27, 28, 39 e 40.

DILL, Ken A et al. The protein folding problem. **Annu. Rev. Biophys.**, Annual Reviews, v. 37, p. 289–316, 2008. Cited 2 times on pages 14 e 23.

DORN, Márcio et al. Three-dimensional protein structure prediction: Methods and computational strategies. **Computational biology and chemistry**, Elsevier, v. 53, p. 251–276, 2014. Cited on page 14.

DREPPER, Ulrich. What every programmer should know about memory. **Red Hat, Inc**, v. 11, p. 2007, 2007. Cited on page 51.

EIBEN, Agoston E; SMITH, James E et al. **Introduction to evolutionary computing**. Berlin: Springer, 2003. v. 53. Cited 4 times on pages 27, 28, 29 e 30.

FACCIOLI, Rodrigo Antonio; BORTOT, Leandro Oliveira; DELBEM, Alexandre CB. Multi-objective evolutionary algorithm nsga-ii for protein structure prediction using structural and energetic properties. **International Journal of Natural Computing Research (IJNCR)**, IGI Global, v. 4, n. 1, p. 43–53, 2014. Cited 2 times on pages 32 e 36.

GAO, Shangce et al. Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 15, n. 4, p. 1365–1378, 2017. Cited 2 times on pages 33 e 36.

GARRET, RH; GRISHAM, CM. Biochemistry 4ed. **University of Virginia, Boston, MA**, 2010. Cited 7 times on pages 14, 18, 19, 20, 21, 22 e 23.

GONÇALVES, José Fernando; RESENDE, Mauricio GC. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, Springer, v. 17, n. 5, p. 487–525, 2011. Cited 4 times on pages 16, 28, 29 e 39.

GU, Jenny; BOURNE, Philip E. **Structural bioinformatics**. Hoboken: John Wiley & Sons, 2009. v. 44. Cited 4 times on pages 14, 15, 22 e 23.

HANDL, Julia; LOVELL, Simon C; KNOWLES, Joshua. Investigations into the effect of multiobjectivization in protein structure prediction. In: SPRINGER. **International Conference on Parallel Problem Solving from Nature**. Berlin, 2008. p. 702–711. Cited 2 times on pages 32 e 35.

JONES, David T. Protein secondary structure prediction based on position-specific scoring matrices. **Journal of molecular biology**, Elsevier, v. 292, n. 2, p. 195–202, 1999. Cited on page 23.

KABSCH, Wolfgang; SANDER, Christian. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. **Biopolymers: Original Research on Biomolecules**, Wiley Online Library, v. 22, n. 12, p. 2577–2637, 1983. Cited on page 39.

KALYANMOY, Deb; DEB, Kalyanmoy. Multi-objective optimization using evolutionary algorithms. **West Sussex, England: John Wiley**, 2001. Cited 5 times on pages 15, 16, 26, 27 e 28.

KNOWLES, Joshua D; CORNE, David W. Approximating the nondominated front using the pareto archived evolution strategy. **Evolutionary computation**, MIT Press, v. 8, n. 2, p. 149–172, 2000. Cited on page 28.

LENA, Pietro Di; NAGATA, Ken; BALDI, Pierre. Deep architectures for protein contact map prediction. **Bioinformatics**, Oxford University Press, v. 28, n. 19, p. 2449–2457, 2012. Cited on page 24.

MA, Jianzhu et al. Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. **Bioinformatics**, Oxford University Press, v. 31, n. 21, p. 3506–3513, 2015. Cited 2 times on pages 24 e 25.

MAIOROV, Vladimir N; CRIPPEN, Gordon M. Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. **Journal of molecular biology**, Elsevier, v. 235, n. 2, p. 625–634, 1994. Cited on page 25.

MARCHI, Felipe; PARPINELLI, Rafael Stubs. A multi-objective approach to the protein structure prediction problem using the biased random-key genetic algorithm. In: IEEE. **2021 IEEE Congress on Evolutionary Computation (CEC)**. New York, 2021. p. 1070–1077. Cited on page 78.

MÁRQUEZ-CHAMORRO, Alfonso E et al. Soft computing methods for the prediction of protein tertiary structures: A survey. **Applied Soft Computing**, Elsevier, v. 35, p. 398–410, 2015. Cited on page 14.

MONASTYRSKYY, Bohdan et al. Evaluation of residue–residue contact prediction in casp10. **Proteins: Structure, Function, and Bioinformatics**, Wiley Online Library, v. 82, p. 138–153, 2014. Cited on page 24.

NARLOCH, Pedro Henrique; KRAUSE, Mathias J; DORN, Márcio. Multi-objective differential evolution algorithms for the protein structure prediction problem. In: IEEE. **2020 IEEE Congress on Evolutionary Computation (CEC)**. New York, 2020. p. 1–8. Cited 3 times on pages 33, 37 e 63.

OLSON, Brian; SHEHU, Amarda. Multi-objective stochastic search for sampling local minima in the protein energy surface. In: **Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics**. New York, NY, USA: Association for Computing Machinery, 2013. (BCB'13), p. 430–439. ISBN 9781450324342. Disponível em: <https://doi.org/10.1145/2506583.2506590>. Cited 2 times on pages 32 e 35.

PARPINELLI, Rafael et al. A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms. **International Journal of Bio-Inspired Computation**, v. 13, p. 1–20, 2019. Cited on page 30.

ROCHA, Gregório Kappaun et al. A multiobjective approach for protein structure prediction using a steady-state genetic algorithm with phenotypic crowding. In: IEEE. **2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)**. New York, 2015. p. 1–8. Cited 2 times on pages 32 e 36.

ROCHA, Gregório K et al. Inserting co-evolution information from contact maps into a multiobjective genetic algorithm for protein structure prediction. In: IEEE. **2018 IEEE Congress on Evolutionary Computation (CEC)**. New York, 2018. p. 1–8. Cited 2 times on pages 33 e 37.

ROHL, Carol A et al. Protein structure prediction using rosetta. **Methods in enzymology**, Elsevier, v. 383, p. 66–93, 2004. Cited 4 times on pages 26, 38, 39 e 63.

SILVA, Renan S; PARPINELLI, Rafael Stubs. A self-adaptive differential evolution with fragment insertion for the protein structure prediction problem. In: SPRINGER. **International Workshop on Hybrid Metaheuristics**. [S.l.], 2019. p. 136–149. Cited on page 14.

SONG, Shuangbao et al. Aimoes: Archive information assisted multi-objective evolutionary strategy for ab initio protein structure prediction. **Knowledge-Based Systems**, Elsevier, v. 146, p. 58–72, 2018. Cited 2 times on pages 33 e 36.

SONG, Shuangbao et al. Adoption of an improved pso to explore a compound multi-objective energy function in protein structure prediction. **Applied Soft Computing**, Elsevier, v. 72, p. 539–551, 2018. Cited 3 times on pages 33, 36 e 63.

TUDELA, José Carlos Calvo; LOPERA, Julio Ortega. Parallel protein structure prediction by multiobjective optimization. In: IEEE. **2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing**. New York, 2009. p. 268–275. Cited 2 times on pages 32 e 35.

VENSKE, Sandra M et al. Ademo/d: An adaptive differential evolution for protein structure prediction problem. **Expert Systems with Applications**, Elsevier, v. 56, p. 209–226, 2016. Cited 2 times on pages 33 e 36.

WILL, Nilcimar Neitzel; PARPINELLI, Rafael Stubs. Comparing best and quota fragment picker protocols applied to protein structure prediction. In: **HIS**. [S.l.: s.n.], 2020. p. 669–678. Cited on page 14.

ZAMAN, Ahmed Bin; PARTHASARATHY, Prasanna Venkatesh; SHEHU, Amarda. Using sequence-predicted contacts to guide template-free protein structure prediction. In: **Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics**. New York, NY, USA: Association for Computing Machinery, 2019. (BCB '19), p. 154–160. ISBN 9781450366663. Disponível em: <https://doi.org/10.1145/3307339.3342175>. Cited 2 times on pages 33 e 37.

ZEMLA, Adam. Lga: a method for finding 3d similarities in protein structures. **Nucleic acids research**, Oxford University Press, v. 31, n. 13, p. 3370–3374, 2003. Cited on page 26.

ZHANG, Gui-Jun et al. Secondary structure and contact guided differential evolution for protein structure prediction. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 17, n. 3, p. 1068–1081, 2018. Cited on page 63.

ZHANG, Jingfen; XU, Dong. Fast algorithm for population-based protein structural model analysis. **Proteomics**, Wiley Online Library, v. 13, n. 2, p. 221–229, 2013. Cited 4 times on pages 16, 30, 31 e 45.

# APPENDIX A – ANOVA P-VALUE

Table 12 – ANOVA p-value for the MO-BRKGA + PC (best solution), considering the GDT. Results are statistically different if the value is greater than 0.05 (in bold on the table).

| **Protein** | **MO-BRKGA Best** | **MO-BRKGA MUFOLD-CL** | **MO-BRKGA + PC MUFOLD-CL** |
|---|---|---|---|
| 1AB1 | **0.14** | 0.00 | 0.00 |
| 1ACW | **0.25** | 0.00 | 0.00 |
| 1AIL | **0.92** | 0.00 | 0.00 |
| 1ALY | **0.07** | 0.00 | 0.00 |
| 1BDD | **0.05** | 0.00 | 0.00 |
| 1CRN | **0.54** | 0.00 | 0.00 |
| 1DFN | **0.05** | 0.00 | 0.00 |
| 1ENH | **0.07** | 0.00 | 0.00 |
| 1GB1 | **0.79** | 0.00 | 0.02 |
| 1HHP | **0.85** | 0.00 | 0.01 |
| 1I6C | **0.61** | 0.00 | 0.00 |
| 1ROP | **0.06** | 0.00 | 0.00 |
| 1ZDD | **0.47** | 0.00 | 0.00 |
| 2KDL | **0.07** | 0.00 | 0.00 |
| 2MR9 | **0.08** | 0.00 | 0.00 |
| 2P81 | **0.61** | 0.00 | 0.00 |
| T0868 | **0.46** | 0.00 | 0.00 |
| T0900 | **0.11** | 0.00 | 0.00 |
| T0968S1 | **0.72** | 0.00 | 0.00 |
| T1010 | **0.05** | 0.00 | 0.00 |

Source: Author

Table 13 – ANOVA p-value for the MO-BRKGA + PC (MUFOLD-CL solution), considering the GDT. Results are statistically different if the value is greater than 0.05 (in bold on the table).

| **Protein** | **MO-BRKGA Best** | **MO-BRKGA MUFOLD-CL** | **MO-BRKGA + PC Best** |
|---|---|---|---|
| 1AB1 | 0.00 | **0.76** | 0.00 |
| 1ACW | 0.00 | 0.00 | 0.00 |
| 1AIL | 0.00 | **0.54** | 0.00 |
| 1ALY | 0.01 | 0.00 | 0.00 |
| 1BDD | 0.00 | **0.89** | 0.00 |
| 1CRN | 0.00 | **0.22** | 0.00 |
| 1DFN | 0.02 | 0.01 | 0.00 |
| 1ENH | 0.00 | **0.11** | 0.00 |
| 1GB1 | 0.01 | **0.18** | 0.02 |
| 1HHP | 0.01 | **0.31** | 0.01 |
| 1I6C | 0.00 | **0.27** | 0.00 |
| 1ROP | 0.00 | **0.28** | 0.00 |
| 1ZDD | 0.01 | **0.06** | 0.00 |
| 2KDL | 0.00 | **0.37** | 0.00 |
| 2MR9 | 0.00 | **0.65** | 0.00 |
| 2P81 | 0.00 | **0.05** | 0.00 |
| T0868 | 0.00 | **0.08** | 0.00 |
| T0900 | 0.00 | 0.03 | 0.00 |
| T0968S1 | 0.00 | **0.56** | 0.00 |
| T1010 | 0.00 | 0.00 | 0.00 |

Source: Author

Table 14 – ANOVA p-value for the MO-BRKGA with MUFOLD-CL, considering the RMSD. Results are statistically different if the value is greater than 0.05 (in bold on the table).

| **Protein** | **NSGA2** | **GDE3** | **DEMO** | **SCDE** | **Rosetta** | **MO-BRKGA Best** | **MO-BRKGA Mean** |
|---|---|---|---|---|---|---|---|
| 1AB1 | - | - | - | - | 0.00 | 0.00 | **0.07** |
| 1ACW | 0.00 | 0.00 | 0.00 | - | **0.53** | 0.00 | **0.10** |
| 1AIL | 0.00 | **0.36** | **0.98** | 0.00 | 0.00 | 0.00 | **0.31** |
| 1ALY | - | - | - | 0.00 | 0.02 | 0.00 | 0.00 |
| 1BDD | - | - | - | - | 0.00 | 0.00 | **0.18** |
| 1CRN | 0.00 | 0.00 | 0.00 | - | 0.00 | 0.00 | **0.29** |
| 1DFN | - | - | - | - | 0.00 | 0.00 | **0.14** |
| 1ENH | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.25** |
| 1GB1 | - | - | - | 0.02 | **0.97** | 0.02 | **0.05** |
| 1HHP | - | - | - | 0.01 | 0.00 | 0.01 | 0.00 |
| 1I6C | - | - | - | 0.00 | 0.00 | 0.00 | **0.77** |
| 1ROP | 0.00 | 0.00 | 0.00 | - | 0.00 | 0.00 | **0.14** |
| 1ZDD | 0.00 | 0.00 | 0.00 | - | 0.00 | 0.00 | **0.93** |
| 2KDL | - | - | - | - | 0.00 | 0.00 | **0.45** |
| 2MR9 | 0.00 | 0.00 | 0.00 | - | 0.00 | 0.00 | **0.99** |
| 2P81 | **1.00** | 0.00 | 0.00 | - | 0.00 | 0.00 | **0.32** |
| T0868 | - | - | - | - | 0.00 | 0.00 | 0.00 |
| T0900 | - | - | - | - | 0.00 | 0.00 | 0.00 |
| T0968S1 | - | - | - | - | 0.00 | 0.00 | 0.00 |
| T1010 | - | - | - | - | **0.06** | 0.00 | 0.00 |

Source: Author

Table 15 – ANOVA p-value for the MO-BRKGA with MUFOLD-CL, considering the GDT_TS. Results are statistically different if the value is greater than 0.05 (in bold on the table).

| **Protein** | **NSGA2** | **GDE3** | **DEMO** | **Rosetta** | **MO-BRKGA Best** | **MO-BRKGA Mean** |
|---|---|---|---|---|---|---|
| 1AB1 | - | - | - | 0.00 | 0.00 | 0.03 |
| 1ACW | **0.27** | 0.01 | **0.13** | **0.27** | 0.00 | **0.30** |
| 1AIL | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.79** |
| 1ALY | - | - | - | 0.00 | 0.00 | **0.09** |
| 1BDD | - | - | - | 0.00 | 0.00 | **0.23** |
| 1CRN | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.08** |
| 1DFN | - | - | - | 0.00 | 0.00 | **0.68** |
| 1ENH | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.20** |
| 1GB1 | - | - | - | **0.42** | 0.00 | 0.04 |
| 1HHP | - | - | - | 0.00 | 0.00 | **0.07** |
| 1I6C | - | - | - | 0.00 | 0.00 | **0.16** |
| 1ROP | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.97** |
| 1ZDD | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.83** |
| 2KDL | - | - | - | 0.00 | 0.00 | **0.06** |
| 2MR9 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | **0.79** |
| 2P81 | 0.00 | **0.75** | **0.07** | **0.60** | 0.00 | 0.00 |
| T0868 | - | - | - | 0.00 | 0.00 | **0.05** |
| T0900 | - | - | - | 0.00 | 0.00 | **0.19** |
| T0968S1 | - | - | - | 0.00 | 0.00 | 0.00 |
| T1010 | - | - | - | 0.00 | 0.00 | **0.09** |

Source: Author