A Multistage Simulated Annealing for Protein Structure Prediction Using Rosetta

Renan S. Silva, Rafael Stubs Parpinelli

[1]Graduate Program in Applied Computing
State University of Santa Catarina (UDESC) – Joinville, SC – Brazil

renan.silva@edu.udesc.br, rafael.parpinelli@udesc.br

Abstract. *The Protein Structure Prediction problem is currently one of the most challenging open problems in Bioinformatics being a NP-Complete problem. In this work, a Multistage Simulated Annealing (MSA) employing different levels of detail for the potential energy function is applied using the Rosetta framework. The backbone and centroid coordinates model is employed being the side chains repacked at the end of the process. Experiments were conducted using four well-known proteins with different degrees of complexity, namely: 1ZDD; 1CRN; 1ENH; 1AIL. The results obtained showed that MSA is able to find better energy function values in all four proteins, and better RMSD in three of them.*

1. Introduction

Proteins are macromolecules that several metabolic, structural and hormonal roles are played them. The function of a protein is directly related to its three dimensional conformation. Thus, by knowing the protein conformation can give insights on the roles that a protein has in a organism, on the design of new drugs, and the better understanding of diseases [Walsh 2002]. Each protein has a unique amino acid sequence, which can be used to identify it. The process of determining a protein sequence, called protein sequencing, is relatively cheap and very reliable. On the other hand, the process of determining the protein structure native conformation usually involves x-ray crystallography or nuclear magnetic resonance that are slow, error prone, and very expensive [Drenth 2007].

Computational modeling of proteins to determine their native structure conformations is known as Protein Structure Prediction and currently it is considered an open problem in computer science and bioinformatics [Dorn et al. 2014]. The Protein Structure Prediction (PSP) problem can be approached with different levels of abstraction in which several of its models are considered to be NP-Complete problems [Guyeux et al. 2014]. Given the complexity of the energy landscape and the number of potential protein conformations make infeasible the use of exact methods for solving the PSP problem. Thus, the use of heuristics and metaheuristics become essential to handle the problem in a feasible time. Some related work using Genetic Algorithms [Borguesan et al. 2015], Memetic Algorithms [Garza-Fabre et al. 2016], Differential Evolution [Narloch and Parpinelli 2017], and hybrid methods [Zhang et al. 2010] have been explored to solve the PSP problem.

The main goal of this work is to explore the PSP problem conformational space at different levels of detail, starting with a coarse grained model and finishing with an all atom configuration. The Rosetta framework provides the potential energy functions and other problem specific routines, and the Simulated Annealing algorithm is employed as optimizer to explore the conformational space considering the backbone and centroid coordinates model. At the end of the optimization process the side chains are repacked to provide an all atom conformation. It is expected that the increasing levels of detail and the use of more information about the search space will lead to better results.

The paper is structured as follows: Section 2 presents the theoretical background relevant to this work, Section 3 describes the proposed method, Section 4 details the experimentation, Section 5 shows the results obtained, and Section 6 presents the conclusions and future work.

2. Theoretical Background

This section describes the theories and concepts that are relevant for a better understanding of this work.

2.1. The Protein Structure Prediction Problem

A protein can have its structure analyzed at different levels. The first one is the primary structure, which consists of the unique sequence that defines a protein. The secondary structure consists of angle patterns that repeat themselves in the protein, forming regular shapes. The most common secondary structures are α-helices (helicoidal shapes), β-sheets (planar shapes), and coils. The tertiary structure corresponds to the three dimensional structure of the protein, and it is also called the native conformation. The quaternary structure are super structures composed of 2 or more proteins. Hence, the PSP problem consists of finding the tertiary structure having the primary structure as input.

There are several protein models that can be utilized to computationally represent a protein. They can be divided into two major classes: lattice and off-lattice models. The lattice models bind the amino acids to a grid of points. This model is the simplest one and can be approached using exact methods [Nunes et al. 2016]. The off-lattice models are able to better represent the proteins since they have more degrees of freedom in space. The most common off-lattice models are: C_α Coordinates, all heavy atoms coordinates, backbone and centroid coordinates, backbone and side chain torsion angles, and all atoms coordinates. The present work employs the backbone and centroid coordinates model and is illustrated in Figure 1. Each amino acid contains 3 dihedral torsion angles for the backbone: ϕ, ψ and ω. The angles ϕ and ψ can assume any value in the range of $[-180, 180]$. The angle ω is most of the time close to 180 due to the planarity of the peptide bond but it is possible to be at 0 degrees when the bond assumes a *cis* configuration. The side chain R_{group} is amino-dependent and is clustered in a small region based on the amino acid type and the secondary structure, representing a centroid.

The process of predicting the native conformation of a protein using only the primary structure as input is known as *ab initio* approach. When some other problem information, e.g. secondary structure information, rotamers, and fragments, are added to the search process then we have a *de novo* approach. Both methodologies are guided by potential energy functions, e.g. AMBER, CHARMM, and Rosetta [Dorn et al. 2014], that are computable approximations of real native conformations. This makes the PSP problem very challenging and encourages the use of problem specific information.

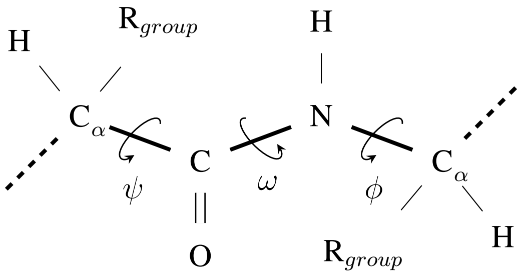

Figure 1. Backbone protein representation

2.2. Rosetta Framework

Rosetta is a software package maintained by RosettaCommons [1]. It is composed of several protocols and methods for working with proteins and other macromolecules, and it has been successfully utilized in the CASP challenges [Ovchinnikov et al. 2017]. Methods to measure and score proteins are provided, as well as means of manipulating the protein structures. All of which is available through an accessible and open source API in C++ and Python.

One of the features that Rosetta provides is a set of scoring functions with different levels of detail. In Figures 2 and 3, a slice of the multidimensional search space is shown for two different energy functions. The image represents the value of the energy function when all protein angles are fixed and only the ϕ and ψ angles of residue 23 from the "1crn" protein is changed. It is possible to see that both functions have the optimal point next to each other and that both energy landscapes have similar features. The function utilized in Figure 2 is referenced in rosetta as "score3" and it only considers the atoms in the backbone of the protein and a centroid as replacement for the side chain. The function utilized for Figure 3 is referenced in rosetta as "ref2015"(Rosetta Energy Function) and it considers all atoms in the protein. The plateau around the optimal point shown in Figure 2 appears due to the lack of the side chain. Since the all atom function has the side chain atoms attached to the side of the back bone whereas the centroid model has not, in the all atom function the close proximity is penalized and in the centroid model it is not (because there is no side chain).

Some of the protocols implemented with Rosetta utilized a sequence of scoring functions. At each stage of the protocol the level of detail is increased. As discussed in [Kmiecik et al. 2016], the use of coarse-grained functions at the start of the process allows a better exploration of the search space, since the energy landscape is more smooth and faster to evaluate. Then, at latter stages, a more detailed energy can be used to find a better description for the protein.

2.3. Simulated Annealing

Simulated Annealing (SA) is a metaheuristic that uses the concept of annealing as a metaphor to guide the optimization process [Kirkpatrick et al. 1983]. Compared to other metaheuristics it is relatively simple and require only few parameters. They are: an initial

[1] https://www.rosettacommons.org/about

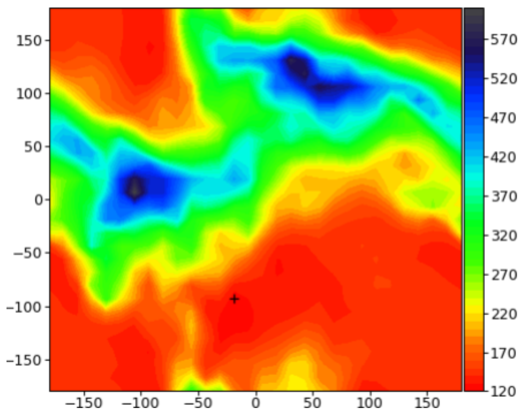

Figure 2. Centroid Function

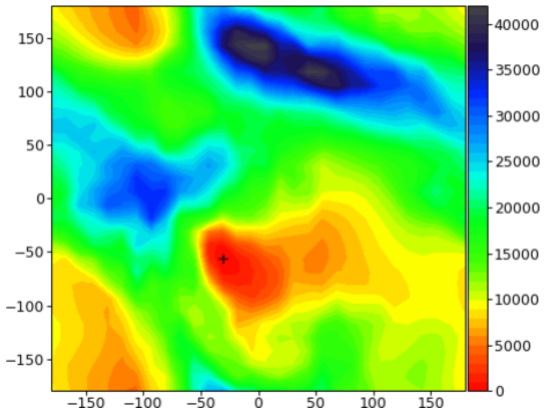

Figure 3. All Atoms Function

temperature (t_0), a final temperature (t_n), a cooling scheme, and the number of iterations (n). The temperature is used to control the chance of accepting a worse solution than the current one. The lower the temperature the lower the chance of accepting a worse solution will be. An increase in the energy (objective function) also decreases the chance of accepting a solution. A *RandomNeighbor(S)* function is needed to generate a random solution S_{new} similar to S. An energy function $E(S)$ is used to evaluate the current solution S. For the PSP problem the energy function E corresponds to the protein scoring function and S is the set of angles describing a protein. S_{best} represents the best set of angles found by SA. The SA pseudo-code is shown in Algorithm 1.

Algorithm 1 Simulated Annealing Pseudo-code

```
$S \leftarrow S_0$
$S_{best} \leftarrow S$
$i \leftarrow 0$
$t_0 \leftarrow InitialTemperature$
$t_n \leftarrow FinalTemperature$
for $i \leq n$ do
    $S_{new} \leftarrow RandomNeighbor(S)$
    $\Delta e \leftarrow E(S_{new}) - E(S)$          ▷ E(.) corresponds to the objective function
    if $E(S_{new}) < E(S)$ then                      ▷ Minimization problem
        $S \leftarrow S_{new}$
        if $E(S) < E(S_{best})$ then
            $S_{best} \leftarrow S$
    else
        if $e^{-\frac{\Delta e}{T}} > Random[0, 1]$ then
            $S \leftarrow S_{new}$
    $i \leftarrow i + 1$
    $T \leftarrow Temperature(t_0, t_n)$
return $S_{best}$
```

3. The Multistage Simulated Annealing Method

The Multistage Simulated Annealing (MSA) makes use of different energy functions from the Rosetta framework, in an increasing order of detail. The main concept of MSA is to use separate stages for each energy function while keeping the same conformation between stage transitions. The main hypothesis is that when optimizing smooth surfaces of simpler energy functions it may lead to promising search regions in more detailed ones. The sampling of the search space is conducted with fragments of size 3 and 9 generated with Rosetta using the predicted secondary structures from PSSpred[2] software.

A simple representation of MSA is available in Figure 4 in which each line represents a stage. Each stage is composed by the optimization algorithm (in red), the problem specific operators (in blue), and the energy functions employed (in yellow), respectively. The proposed MSA is composed by 6 different stages, where the first 5 uses a Simulated Annealing to optimize the backbone angles and the last one is used exclusively for repacking the side chains into the model. The first three stages are responsible for global search and the following two stages are responsible for local search, as described next.

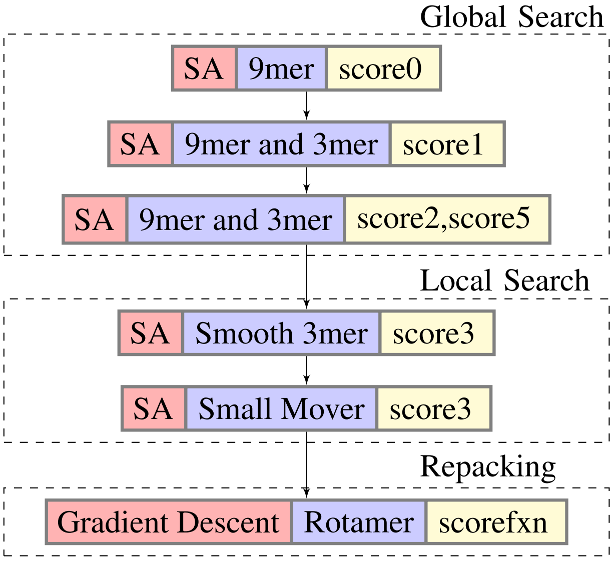

Figure 4. For each stage, red indicates the optimization algorithm, blue indicates the operators employed, and yellow indicates the energy function used

The first stage of MSA utilizes an energy function that considers only the repulsive force of the side chain centroids. Along with the use of fragments of size 9 (9-mer), this stage provides a quick sampling of the conformational space with the goal of finding a good starting point for the next stages. The aim of this stage is to find a conformation where its parts does not intersect itself (a self avoiding walk), which can be verified by checking if the energy function, namely “score0”, has its value set to 0.

The second and third stages are responsible for exploring the protein conformational space in a more broad way. Hence, fragments of size 9 and 3 are added at each SA iteration. The energy function “score1” is used on stage two, where it consider the *van der Waals* interactions, the formation of secondary structures and disulfide bonds. On stage

[2] https://zhanglab.ccmb.med.umich.edu/PSSpred/

three, the energy functions “score2” and “score5” are used in an interchanging way. The process starts with “score2” and, after a fixed amount of function evaluations, the energy function in use is changed to “score 5” that is employed for the same amount of function evaluations. This loop is repeated until the maximum number of function evaluations for this stage is reached. The functions “score2” and “score5” are in the same stage because they are very similar. The main difference is that some weights are changed in Rosetta framework and it starts to consider the overall compactness of the conformation. The aim of switching between these two functions is to allow the scape of local minima by using two similar energy functions with different set of weights.

Stages four and five are dedicated to local search. Both stages use “score3” energy function which adds more terms related to compactness, solvent accessibility and positioning of side chain centroids. At the begging of stage four, for a small amount of iterations, fragments of size 3 are used. Then, for the rest of this stage, two fragments of size 3 are added at once. The first one is randomly assigned to a position and the second one is placed where it will minimize the disruption caused by the placement of the first. This strategy increases the acceptance rate of the perturbations made without destroying the conformation constructed so far and is called “Smooth” in Rosetta framework. Stage 5 operates by doing random changes to the dihedral angles of a randomly chosen amino acid. The angle changes are to be very small, and if the change is attempted on an α-helix or a β-sheet than the perturbation is even smaller. This allows for a final refinement step without removing the shape of the secondary structures found and is called in Rosetta by “Small Mover”.

The last stage (repacking) adds the side chains based on a rotamer database [Dunbrack and Karplus 1993]. The repacking procedure consists of adding in random order the side chains with the goal of minimize the potential energy (“scorefxn”). Hence, only the terms dependent on the side chain need to be recalculated. This allows the system to perform a full atom prediction for the target protein.

4. Experiments Setup

To evaluate the proposed method 4 well known proteins were used as target for predictions: 1ZDD, 1CRN, 1ENG, and 1AIL. All 4 proteins are available at PDB [Berman et al. 2002]. Their size, number of angles to optimize, and secondary structures are listed in Table 1. They have an increasing number of angles to optimize, and the 1CRN protein has the presence of α-helix and β-sheet secondary structures being the hardest one to optimize.

During the optimization process the main goal is to minimize the scoring function. However, since the scoring function is an approximation of the physical interactions and

Name	Size	Backbone Angles	Structure
1ZDD	35	105	α
1CRN	46	138	α, β
1ENH	54	162	α
1AIL	72	216	α

Table 1. The target proteins

its surface is very rugged and highly multimodal, the optimal point may not correspond to the protein native conformation. With the goal of validating the final predicted protein, the distance between the prediction and the native conformation is measured. This allows the assessment of how far from the ideal the prediction is and it also allows to compare between different related work. The distance measure is shown in Equation 1.

$$RMSD_{\alpha}(A,B) = \sqrt[2]{\frac{\sum_{i=n}^{n}(A_i - B_i)^2}{n}} \tag{1}$$

RMSD stands for Root Mean Squared Deviation. The RMSD is the average of the squared deviation between two structures. In this case, the α-Carbon of the predicted conformation and the native one are compared. A value of 0 means that the two conformations are identical.

The parameters used during the tests are shown in Table 2, and Table 3. All parameters were set empirically meaning that no special adjustment was performed. Hence, this is pointed as future work. Table 2 shows the temperature schedules, and the initial (t_0) and final (t_n) temperatures for each stage. The temperature schedules ts_1 and ts_2 are presented in Equations (2) and (3), respectively. Table 3 shows the amount of function evaluations performed in each stage, as well as the total amount of function evaluations for each run. For stage3, “score2” and “score5” functions are interchanged at each 10000 function evaluations.

	Values
stage1	$ts_1, t_0 = 5.0, t_n = 0.5$
stage2	$ts_1, t_0 = 5.0, t_n = 0.5$
stage3	$ts_1, t_0 = 5.0, t_n = 0.5$
stage4	$ts_2, t_0 = 5.0, t_n = 0.0$
stage5	$ts_2, t_0 = 1.0, t_n = 0.2$

Table 2. SA temperature configuration

	Function Evaluations
stage1	10000
stage2	10000
stage3	100000
stage4	80000
stage5	300000
Total	500000

Table 3. Function evaluations

$$ts_1(i) = \frac{t_0 - t_n}{cosh(5.0 * \frac{i}{n})} + t_0 \tag{2}$$

$$ts_2(i) = \frac{t_0 - t_n}{n} * i + t_0 \tag{3}$$

For each protein, 10 runs were executed. The tests were run on a machine equipped with an Intel® Core™ i5-3570k clocked at 4.2GHz, 16GB of RAM clocked at 1400MHz, and running a GNU/Linux operating system.

The results obtained by our method are compared with the results obtained by two well known algorithms, namely, Differential Evolution (DE) and Genetic Algorithms (GA), reported in [Narloch and Parpinelli 2017] and [Borguesan et al. 2015], respectively. [Narloch and Parpinelli 2017] presents a study where different mutation operators are applied sequentially, one after another during a run. The approach is called DE Cascade and two sequences of mutation operators are explored (DE$_{C1}$ and DE$_{C2}$). In [Borguesan et al. 2015], a Genetic Algorithm using heterogeneous population, diversity control and operators that considers the experimental dihedral angle distribution are utilized. These works were chosen because they use the same set of proteins, the same energy function framework and the same representative model for optimization.

5. Results and Analysis

Table 4 shows the results obtained. First column indicates the proteins, second column shows the algorithms, third column shows the overall lowest energy values obtained by each algorithm, fourth column indicates the α-Carbon RMSD for the lowest energy, and the last column indicates the average and standard deviations for all 10 run. Bold cells indicate best results.

When comparing the overall lowest energy values, the MSA was able to outperform the other methods in all 4 target proteins. This indicates that, for this particular set of proteins, the proposed approach is a better optimizer than the other methods considered. Considering the RMSD for the conformation with the lowest energy value, the MSA was able to outperform the other approaches in 3 out of 4 target proteins. For 1CRN the GA obtained a better RMSD. Ideally, there would be a direct relation between the RMSD and the energy value. However, since the energy functions are computational approximations of real work conformations and the search space is highly multimodal, there may be situations where a low energy value may have a higher RMSD than another conformation with a higher energy. Analyzing the average and standard deviation of the results obtained, fifth column, it is possible to identify that MSA is a significantly better optimizer. Statistical tests were not performed because other results were obtained directly from respective original articles. However, it is clear that there is no standard deviations overlapping among the MSA and other approaches.

Protein	Version	Min. Energy	RMSDα(Å)	Avg. Energy
1ZDD	DE$_{C_1}$	54.27	7.67	82.97 ± 15.49
	DE$_{C_2}$	65.77	9.42	82.76 ± 9.22
	GA	-40.40	10.9	-36.20 ± 2.60
	MSA	**-62.99**	**2.62**	$\mathbf{-48.96 \pm 7.77}$
1CRN	DE$_{C_1}$	82.86	21.56	126.95 ± 25.98
	DE$_{C_2}$	72.48	15.44	109.08 ± 22.96
	GA	-22.70	**5.8**	-18.20 ± 2.9
	MSA	**-76.93**	6.96	$\mathbf{-54.01 \pm 17.30}$
1ENH	DE$_{C_1}$	294.25	14.72	372.11 ± 52.05
	DE$_{C_2}$	255.54	19.28	320.38 ± 41.06
	GA	-56.08	14.99	-51.52 ± 1.94
	MSA	**-95.86**	**5.70**	$\mathbf{-80.75 \pm 8.48}$
1AIL	DE$_{C_1}$	357.84	25.00	440.63 ± 58.11
	DE$_{C_2}$	332.54	16.88	411.81 ± 56.84
	GA	-75.07	12.34	-71.08 ± 3.35
	MSA	**-128.55**	**8.27**	$\mathbf{-117.54 \pm 10.28}$

Table 4. Results obtained

Table 5 shows the average processing time for each protein measured in minutes. For the DE approaches the reported processing times were 48 minutes for the smallest protein (1ZDD), and 2 hours and 52 minutes for the largest one (1AIL). The reported processing time for the GA was about 12 hours for each run for the same set of proteins, neither limiting the number of generations nor the maximum number of function evaluations. Even though the hardware utilized are not the same, the time difference can not be explained just by the hardware difference but also by the simplicity of the proposed method. The processing times achieved by the MSA are expressive when compared with other approaches showing its robustness.

Protein	Time (min)
1ZDD	3.5 ± 0.17
1CRN	5.57 ± 0.22
1ENH	6.78 ± 0.37
1AIL	9.56 ± 0.01

Table 5. Processing time per protein

In Figure 5(a-d) the predicted proteins (in green) are compared to their native conformations (in red). For 1ZDD and 1AIL the secondary structures were correctly assembled. 1ZDD has a near native conformation found, where only the middle coil section was misplaced. For 1CRN only one of the two α-helices were found and none of the β-sheets. For 1EHN, two out of tree α-helices were found. The third one was misfolded as a β-bridge. The two α-helices found were close to the native conformation, but the other one was miss oriented. For 1AIL, all secondary structure was found. One of the α-helices was shorter than the native conformation and it folded in the wrong side of the protein.

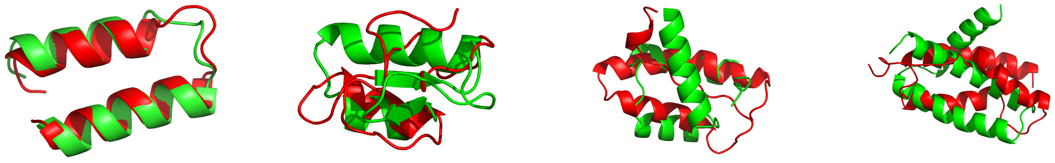
(a) 1ZDD (b) 1CRN (c) 1ENH (d) 1AIL

Figure 5. Predicted Proteins (green) compared to the Native conformations (red)

6. Conclusions

This work presented a Multistage Simulated Annealing Algorithm to solve the Protein Structure Prediction problem. The proposed approach applied different energy functions in an increasing level of detail using Rosetta framework. Also, non-homologous fragment insertion and a final repacking stage were employed.

Based on the results obtained we observed that the use of different levels of detail leads to a better overall prediction. The MSA is able to construct a solution with a better angle distribution resembling the native conformation by using fragment insertion. Also, the sequence of stages using the backbone and centroid coordinates model reduces the complexity of the problem providing a reliable distribution of angles for the side chain repacking stage.

As future works it is possible to better adjust the set of parameters of MSA, also considering the application of parameter control techniques. To include more and larger proteins in the experimentation set may provide a better insight on MSA's efficiency.

References

Berman, H. M., Battistuz, T., Bhat, T., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., et al. (2002). The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907.

Borguesan, B., e Silva, M. B., Grisci, B., Inostroza-Ponta, M., and Dorn, M. (2015). Apl: An angle probability list to improve knowledge-based metaheuristics for the three-dimensional protein structure prediction. *Computational biology and chemistry*, 59:142–157.

Dorn, M., e Silva, M. B., Buriol, L. S., and Lamb, L. C. (2014). Three-dimensional protein structure prediction: Methods and computational strategies. *Computational biology and chemistry*, 53:251–276.

Drenth, J. (2007). *Principles of protein X-ray crystallography*. Springer Science & Business Media.

Dunbrack, R. L. and Karplus, M. (1993). Backbone-dependent rotamer library for proteins application to side-chain prediction. *Journal of molecular biology*, 230(2):543–574.

Garza-Fabre, M., Kandathil, S. M., Handl, J., Knowles, J., and Lovell, S. C. (2016). Generating, maintaining, and exploiting diversity in a memetic algorithm for protein structure prediction. *Evolutionary computation*, 24(4):577–607.

Guyeux, C., Côté, N. M.-L., Bahi, J. M., and Bienia, W. (2014). Is protein folding problem really a np-complete one? first investigations. *Journal of bioinformatics and computational biology*, 12(01):1350017.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A. E., and Kolinski, A. (2016). Coarse-grained protein models and their applications. *Chem. Rev*, 116(14):7898–7936.

Narloch, P. H. and Parpinelli, R. S. (2017). The protein structure prediction problem approached by a cascade differential evolution algorithm using rosetta. In *6th Brazilian Conference on Intelligent Systems*, pages 294–299.

Nunes, L. F., Galvão, L. C., Lopes, H. S., Moscato, P., and Berretta, R. (2016). An integer programming model for protein structure prediction using the 3d-hp side chain model. *Discrete Applied Mathematics*, 198:206–214.

Ovchinnikov, S., Park, H., Kim, D., DiMaio, F., and Baker, D. (2017). Protein structure prediction using rosetta in casp12. *Proteins: Structure, Function, and Bioinformatics*.

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

Zhang, X., Wang, T., Luo, H., Yang, J. Y., Deng, Y., Tang, J., and Yang, M. Q. (2010). 3d protein structure prediction with genetic tabu search algorithm. *BMC systems biology*, 4(1):S6.