Sunil Template

Contents

Chapter 1

Multi objective evolutionary algorithms applied to Protein Structure Prediction Problem

Author Name1

Affiliation text1

Author Name2

Affiliation text2

Author Name3

Affiliation text3

Author Name4

Affiliation text4

1.1 Introduction

1.2 Protein Structure Prediction

Proteins are macromolecules made out of twenty different amino acids, also referred to as residues. An amino acid has a peptide backbone and a distinctive side chain group. The peptide bond is defined by an amino group and a carboxyl group connected to an alpha carbon to which a hydrogen and side chain group are attached.

Amino acids are combined to form sequences which are considered the primary structure of the peptides or proteins. The secondary structure is the locally ordered structure brought about via hydrogen bounding mainly within the peptide backbone. The most common secondary structure elements in proteins are the alpha helix and the beta sheet. The tertiary structure is the global folding of a single polypeptide chain.

Under specific conditions, the protein sequence folds into a unique native 3-d structure. Each possible protein fold has associated energy. The *thermodynamic hypothesis* states that the native structure of a protein is the one for which the free energy achieves the global minimum. Based on this hypothesis, many methods that search for the protein native structure define an approximation of the protein energy and use optimization algorithms that look for the protein fold that minimizes this energy. These approaches mainly differ in the type of energy approximation employed and in the characteristics of the protein modeling.

1.2.1 The HP Model

The Protein Structures are very complex. Use a representation close to the real would be impossible for current computers to process the information in a reasonable time. Having this in mind, Lau and Dill [13] created a model called *Hydrophobic-Hydrophilic* Model (HP Model), to represent the proteins using simplifications. The model can be used either to represent proteins in a 2D space or 3D space.

The HP model considers two types of residues: hydrophobic (H) residues and hydrophilic or polar (P) residues. A protein is considered a sequence of these two types of residues, which are located in regular lattice models forming self-avoided paths. Given a pair of residues, they are considered neighbors if they are adjacent either in the chain (connected neighbors) or in the lattice but not connected in the chain (topological neighbors).

The total number of topological neighboring positions in the lattice (z) is called the lattice coordination number.

For the HP model, an energy function that measures the interaction between topological neighbor residues is defined as $\epsilon_{HH} = -1$ and $\epsilon_{HP} = \epsilon_{PP} = 0$. The HP problem consists of finding the solution that minimizes the total energy. In the linear representation of the sequence, hydrophobic residues are

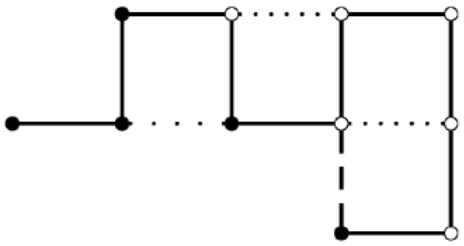

FIGURE 1.1: One possible configuration of sequence $HHHPHPPPPPH$ in the HP model. There is one HH (represented by a dotted line with wide spaces), one HP (represented by a dashed line) and two PP (represented by dotted lines) contacts.

represented with the letter H and polar ones, with P. In the graphical representation, hydrophobic proteins are represented by black beads and polar proteins, by white beads. Figure 1.2.1 shows the graphical representation of a possible configuration for sequence $HHHPHPPPPPH$ in a 2D space. The energy that the HP model associates with this configuration is -1 because there is only one HH contact, arisen between the second and fifth residues.

Different heuristic approaches have been developed to decrease the computational complexity related to the protein structure determination process. Mono and Multi-objective methods have been used, trying to define which methods have better results in the study of the PSP Problem.

Among these approaches, there are studies that explore the protein structure prediction combined with evolutionary algorithms like for example, Lin and Su [15] proposed an mono objective EA (Evolutionary Algorithm) working with a local search strategy for the PSP Problem using the simplified model Hydrophobic-Polar 2D (HP-2D).

In other work, Lin and Su [16] applied a hybrid genetic-based PSO algorithm to HP-3D model with relative representation, optimizing the crossover and mutation operators to improve results in the Protein Folding process. The algorithm was a improved version of a GA based on a PSO, where the solutions were encouraged to move toward their own best solution.

Custódio and Dardenne [6] proposed a Multiple Minima Genetic Algorithm for PSP. The algorithm included a phenotype based crowding mechanism for the maintenance of useful diversity which increase the population's performance and granted the algorithm multiple solutions capabilities.

Brasil *et al.*[21] proposed an multi objective algorithm in tables and compare its performance with the NSGAII [7], optimizing two energy functions, both very important for the the folding process: the van der Walls and Electrostatic functions.

The author of [10] also proposes the using of a multi objective algorithm in

tables, similar to the proposed method by [21], however, using the HP model for the representation and solution evaluation.

1.3 Multi-objective Optimization

Evolutionary Algorithm (EA) is a optimization and search technique, highly parallel, inspired by the Darwinian principle of natural selection and genetic reproduction. The nature principles that inspire the EAs are simple. According to the theory of C, Darwin, the principle of natural selection favors individuals with high fitness, therefore, with high probability of reproduction. Individuals with more descendants have more chance to perpetuate their genetic code in future generations. The genetic codes is what gives the identity of each individual and are represented in the chromosomes. These principles are used in the construction of computational algorithms, that searches for better solutions given a specific problem by the evolution of a population of solutions coded in artificial chromosomes – data structures used to represent a feasible solution for a given problem in the algorithm execution [17].

Real world problems commonly have multiple objectives to minimized/maximized and are present in most knowledge areas. To optimize multi objective problems, are considered two or more objectives witch usually are conflicting. To these problems is impossible to find one unique solution. A set of solutions is reached evaluating the Pareto dominance relation [2] between the solutions. The main objective is to find the solutions that are non-dominated by any other. A solution dominates other, if and only if, it was better in at least one of the objectives, without being worst in any of the objectives. The set of non-dominated solutions constitutes the Pareto Front. Finding the the real Pareto Front is a NP-hard problem [9], this way, the objective is to find a good approximation of this front.

Multi-Objective Evolutionary Algorithms (MOEAS) are extensions of EAs to multi objective problems that applies the concepts of Pareto dominance to create different strategies to evolve and diversify the solutions. In this work were used two MOEAs: NSGAII [7] and IBEA [24].

1.3.1 Non-dominated sorting Genetic Algorithm II

The main characteristic of this algorithm is a strong elitism mechanism, classifying at each generation every solution in different fronts according with the non-dominance relation (line 15 of Algorithm 1). After the classification, solutions from the first front, are non-dominated by any other solution. Solutions from the second front are dominated only by the solutions of the first front, and so on. For solutions of the same front, the algorithm uses a Crowding Distance operator to calculate how distant are the neighbors of a given

solution (line 19 of Algorithm 1). Solutions with high values of Crowding Distance have priority, because they will contribute more to the population's diversity. The binary tournament selects solutions from the small front with the higher values of Crowding Distance. A new population is generated using the crossover and mutation operators (line 25 of Algorithm 1).

Algorithm 1 NSGAII

```
1:  N ← Population Size
2:  T ← Max evaluations
3:  P_0 ← CreatePopulation(N);
4:  CalculateFitness(P_0);
5:  FastNonDominatedSort(P_0);
6:  Q_0 ← 0
7:  while Q_0 < N do
8:      Parents ← BinaryTournament(P_0);
9:      Children ← CrossoverMutation(Parents);
10:     Q_0 ← Children
11: end while
12: CalculateFitness(Q_0);
13: t ← 0
14: while t < T do
15:     R_t ← P_t ∪ Q_t;
16:     Fronts ← FastNonDominatedSort(R_t);
17:     P_{t+1} ← 0
18:     i ← 0
19:     while P_{t+1} + Front_i < N do
20:         CrowdingDistanceAssignment(Front_i);
21:         P_{t+1} ← P_{t+1} ∪ Front_i
22:         i ← i + 1
23:     end while
24:     CrowdingDistanceSort(Front_i);
25:     P_{t+1} ← P_{t+1} ∪ Front_i[1 : (N − P_{t+1})]
26:     Parents ← BinaryTournament(P_{t+1});
27:     Q_{t+1} ← CrossoverMutation(Parents);
28:     t ← t + 1
29: end while
30: return P ← Set of non-dominated solutions.
```

1.3.2 IBEA (Indicator-Based Evolutionary Algorithm)

In the multi-objective optimization context, optimizing consists in find a front with a good approximation to the true Pareto front. However, there is no general definition about what is the true Pareto front. This way, indicators have been used to evaluate the quality of a approximation front. The

hypervolume is a example of indicator to the evaluation and comparison of fronts.

The IBEA is an algorithm that considers the optimization by the use of quality indicators. The indicator is the way used to evaluate the non-dominated set of solutions [8]. To use the IBEA it is necessary define which indicator will be used to associate each ordered pair of solutions to a scalar value. One of the most used indicators is the *hypervolume* due to its capacity of evaluate the convergence and diversity at the same time of the search process [12].

$$F(x_i) = \sum_{x_j \in (P - x_i)} -e^{\frac{-I_{Hy}(x_j, x_i)}{k}} \tag{1.1}$$

For the IBEA fitness calculation (Equation 1.1), k is a parameter commonly used with a value of 0.05. The value for $F(x_i)$ corresponds to a quality loss measure of the approximation to the Pareto front if the solution x_i was removed of the population [8], based on the value of the quality indicator I_{Hy}, in this case, the *hypervolume*. Based on the fitness calculation described above, the basic IBEA algorithm consists in iteratively do the selection (line 10 of Algorithm 2), crossover, mutation (line 11 of Algorithm 2) and environment selection, removing the worst individual from the population and updating the values of fitness of the remaining individuals (lines 4 to 8 of Algorithm 2).

Algorithm 2 IBEA

```
1:  N ← Population Size
2:  T ← Max Evaluations
3:  k ← Scale factor of Fitness
4:  P ← CreatePopulation(N);
5:  m ← 0
6:  CalculateFitness(P);
7:  while m ≥ T or other stop criterion is reached do
8:      P̄ ← BinaryTournament(P);
9:      P ← CrossoverMutation(P̄);
10:     m ← m + 1
11:     while Size(P) > N do
12:         x* ← WorstIndividualByFitness();
13:         RemoveFromPopulation(x*, P);
14:         CalculateFitness(P);
15:     end while
16: end while
17: return P ← Set of non-dominated solutions
```

[1] *Hypervolume*: Proposed quality indicators used in the study of [25], denoted as the "size of the covered search space". This indicator has two important advantages in relation to others [23]: 1 - Sensitive to any kind of improvement in the approximation set in relation

1.4 Proposed method

Two multi-objective approaches were designed in this chapter, using the MOEAs (NSGAII and IBEA) described on subsection 1.3. The relative representation was chosen to represent the chromosomes. Integer vectors are used whereas the genes specifies which direction, relative to the previous residue, should be placed the next residue. The genes can assume only tree values (0,1,2), 0 indicates that next residue should be placed on right of the previous one, 1 indicates that the next residue should be placed on left from the previous and 2 indicates that the next residue should be placed in front of the previous one. Figure 1.2 shows an example of a hypothetical chromosome and the path generated by it in the 2D lattice.

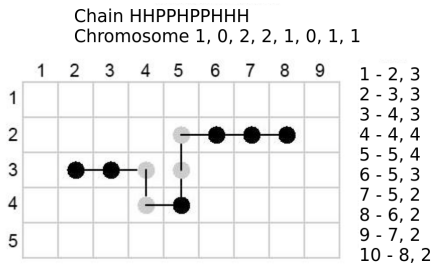

FIGURE 1.2: Example of a conformation generated by a chromosome with relative representation.

The first approach consisted on applying two well-known state of art MOEAs (IBEA and NSGAII) to the PSP using the HP-2D model. The genetic operators used by IBEA and NSGAII algorithms, in this approach, were only the single point crossover in the case of crossover and bit flip mutation for mutation operator. It was decided to use only single point crossover and bit flip mutation because this combination of operators presented better results in previous experiments realized within the PSP problem and the HP-2D model.

In the case of the second approach the IBEA and NSGAII algorithms were modified in order to improve their results when compared with the first approach. The modifications implemented will be described next:

- **Pool of operators**: As mentioned in section X, the use of traditional operators usually does not guide the search to prominent regions of the search space of the HP-2D model. In order to supplement the MOEAs, a pool of operators were designed based on the literature and are presented on table X. For every mating the crossover and mutation operators are

to other set. 2 - As result of 1, the indicator guarantee that for any approximation set A that has high values of hypervolume, also has all the solutions of the true Pareto front.

selected randomly from the pool of operators and than applied. Also the crossover and mutation operators are always applied differently from the first approach which uses a probability of occurrence. The pool of operators will be described next:

- Single Point Crossover (1x): A single point on both parent individuals is selected. All data beyond that point in either individual will be swapped between the two parent individuals. Resulting in two distinct child individuals [11].
- Two Points Crossover (2x): Two points are selected on both parent individuals. Everything between the two points is swapped between the parents. Building two new distinct individuals [11].
- Multi Points Crossover (MPX): The MPX operator is similar to 2X, but the number of points, c, is a function of the sequence length, n, given by $c = int(n \times 0.1)$ [5]. The MPX operator is usually used to promote structural diversity by performing a random shuffle between individuals, although not as uniform crossover.
- Bit Flip Mutation (BFM): The BFM operator selects one random gene from a parent individual and changes it to other value. Resulting in one new individual [11].
- Loop Move Mutation (LMM): The LMM operator swaps the directions between two randomly chosen consecutive genes. This operator introduces a corner movement [3]. Figure 1.4 presents a example of application of this operator.

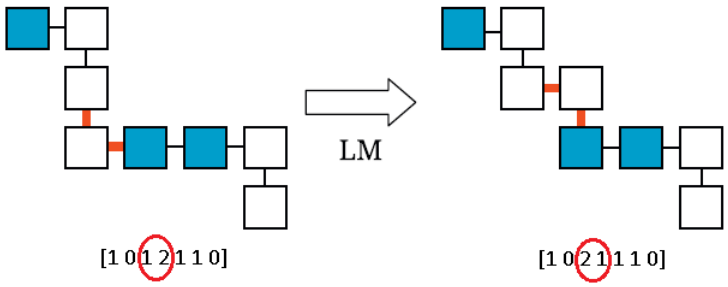

FIGURE 1.3: Example of application of the LMM operator. The genes from the red circle of leftmost figure were swapped resulting in the rightmost figure.

- Loop Move Mutation (LoMM): This operator is similar with LMM however exchanges directions between genes that are five positions apart on the sequence creating a loop movement. Both LMM and LoMM is useful to generate modifications on compact structures [6].

  - Segment Mutation (SM): This operator changes a random number of consecutive genes (from two to seven) into new random directions. This operator introduces large conformational changes and has a high probability of creating collisions, in order to avoid too much invalid solutions the repair mechanism is applied on the generated child [6].
  - Opposite Mutation (OM): This operator changes a random number of consecutive genes to its inverse position. In the case of the relative representation to the HP-2D model, only left and right can be mapped to its inverse.

- **Backtrack Initialization**: Traditionally the initial population of solutions are generated randomly in the presented MOEAs. This have a great potential of generating a large number of invalid solutions for the PSP problem within the HP-2D model. Solutions that are not self-avoiding walk (SAW) are said to contain collisions. If the initial population is fully generated randomly the evolutionary algorithms will spend time processing invalid solutions until getting good results. In order to subdue this problem a backtrack strategy should be applied. In this approach 20 percent of the initial population were generated using the backtrack initialization.

For both approaches the same objectives were used and will be explained at next:

- **Energy value**: This is the main objective and consist in the energy of given protein conformation. The goal is to minimize the energy value and it is calculated as described in section X. This objective guides the search progress towards regions that the energy associate to the protein conformations are minimal. Thus, achieving protein conformations which are closest to native structure of a protein.

- **Distance between the two farthest residues**: This is a secondary objective and it was inspired by the related work X. The motivation for this objective is that more compact conformations tend to have more hydrophobic contacts which means a lesser energy value. The distance between two residues is calculated using the Euclidean distance.

The relative representation it is subject to generation of invalid solutions using the HP-2D model. A solution is considered invalid when the solution does not perform a self-avoiding walk (SAW) as mentioned before. In other words a invalid solution is when a given residue collides with another already placed on the lattice. A simple mechanism for repairing these situations was developed, the code can be seen in the Algorithm 3.

This mechanism was implemented because in previous experiments was observed that the number of infeasible solutions was too big. It is necessary

Algorithm 3 Mechanism to repair infeasible solutions

1. Obtains the direction that next residue should be placed.
2. Verifies if this direction will cause collision.
3. If the a collision is detected, a new direction is used.
4. Repeat the step 2 and 3, until be possible to place the next residue, or if all directions were tested and cause collisions.
5. If was possible to place the next residue, the mechanism achieved success, if not, the solution is considered infeasible and it will be penalized in the evaluation process.

mention that the even with the mechanism to repair solutions, there are still infeasible solutions because the mechanism can not always repair. Thus, infeasible solutions are penalized by subtracting the number of collisions to the quantity of topological neighbors. This mechanism is used by the evaluation process of both approaches described before (IBEA and NSGAII without any major modifications and the same algorithms with the modifications mentioned).

To evaluate and compare the performance of multi-objective algorithms, quality indicators are commonly used. In this study was used the hypervolume indicator, which considers the volume of the search space dominated by the known pareto front [26] of an algorithm. Higher hypervolume value means that the quality is better than one lesser hypervolume value.

1.5 Experiments

This section presents the set of experiments designed to evaluate the performance of the approaches introduced in section 1.4. The HP sequences used in the experiments are shown in table 1.1, those instances have been used in previously works such as [1, 19, 22, 4, 18, 20, 14]. The values presented in table 1.1 correspond to the sequence identifier, the size of aminoacid sequence, the best known solutions ($H(x*)$) for the HP-2D model and the sequence itself.

1.5.1 Comparison between the two approaches

1.6 Conclusion

TABLE 1.1: HP instances used in the experiments. The search space of each instance is 2^n where n is the size of the instance.

inst.	size	$H(\mathbf{x}^*)$	sequence
$sq1$	20	-9	$HPHPPHHPHHPHPHHPPHPH$
$sq2$	24	-9	$HHPPHPPHPPHPPHPPHPPHPPHH$
$sq3$	25	-8	$PPHPPHHP^4HHP^4HHP^4HH$
$sq4$	36	-14	$P^3HHPPHHP^5H^7PPHHP^4HHPPHPP$
$sq5$	48	-23	$PPHPPHHPPHHP^5H^{10}P^6$ $HHPPHHPPHPPH^5$
$sq6$	50	-21	$HHPHPHPHPH^4PHP^3HP^3HP^4$ $HP^3HP^3HPH^4\{PH\}^4H$
$sq7$	60	-36	$PPH^3PH^8P^3H^{10}PHP^3$ $H^{12}P^4H^6PHHPHP$
$sq8$	64	-42	$H^{12}PHPH\{PPHH\}^2PPH\{PPHH\}^2$ $PPH\{PPHH\}^2PPHPHPH^{12}$
$sq9$	85	-53	$H^4P^4H^{12}P^6H^{12}P^3H^{12}P^3$ $H^{12}P^3HP^2H^2P^2H^2P^2HPH$
$sq10$	100	-48	$P^6HPH^2P^5H^3PH^5PH^2P^4H^2$ $P^2H^2PH^5PH^{10}PH^2PH^7$ $P^{11}H^7P^2HPH^3P^6HPH$
$sq11$	100	-50	$P^3H^2P^2H^4P^2H^3PH^2PH^2PH^4$ $P^8H^6P^2H^6P^9HPH^2PH^{11}P^2$ $H^3PH^2PHP^2HPH^3P^6H^3$

TABLE 1.2: My caption

inst	Algorithm	Statistics	Hypervolume	inst	Algorithm	Statistics	Hypervolume
sq1	NSGAII	Avg	**0.742827**	sq5	NSGAII	Avg	0.707654
		Std D	0.106315			Std D	0.082611
		Min	0.391844			Min	0.393669
		Max	0.867612			Max	0.814349
	M_NSGAII	Avg	0.720864		M_NSGAII	Avg	**0.758128**
		Std D	0.131351			Std D	0.062315
		Min	0.296272			Min	0.586954
		Max	0.869180			Max	0.854699
sq2	NSGAII	Avg	0.680572	sq6	NSGAII	Avg	0.667771
		Std D	0.083445			Std D	0.132218
		Min	0.482805			Min	0.386680
		Max	0.861476			Max	0.8436351
	M_NSGAII	Avg	**0.712275**		M_NSGAII	Avg	**0.774017**
		Std D	0.137226			Std D	0.063231
		Min	0.294348			Min	0.678249
		Max	0.864054			Max	0.885620
sq3	NSGAII	Avg	0.605000	sq7	NSGAII	Avg	0.784483
		Std D	0.116475			Std D	0.063257
		Min	0.293469			Min	0.553444
		Max	0.766270			Max	0.848307
	M_NSGAII	Avg	**0.639854**		M_NSGAII	Avg	**0.792843**
		Std D	0.111780			Std D	0.033062
		Min	0.293469			Min	0.701445
		Max	0.766270			Max	0.844117
sq4	NSGAII	Avg	0.702280	sq8	NSGAII	Avg	0.677464
		Std D	0.068983			Std D	0.041287
		Min	0.569481			Min	0.580519
		Max	0.818438			Max	0.768068
	M_NSGAII	Avg	**0.740153**		M_NSGAII	Avg	**0.705798**
		Std D	0.075271			Std D	0.053048
		Min	0.569481			Min	0.600804
		Max	0.819282			Max	0.831057

TABLE 1.3: My caption

inst	Algorithm	Statistics	Hypervolume	inst	Algorithm	Statistics	Hypervolume
sq1	IBEA	Avg	**0.786571**	sq5	IBEA	Avg	0.733464
		Std D	0.067660			Std D	0.128757
		Min	0.674848			Min	0.355967
		Max	0.867089			Max	0.892926
	M_IBEA	Avg	**0.786571**		M_IBEA	Avg	0.807637
		Std D	0.099424			Std D	0.039620
		Min	0.391844			Min	0.701016
		Max	0.866044			Max	0.891029
sq2	IBEA	Avg	0.719960	sq6	IBEA	Avg	0.728699
		Std D	0.080727			Std D	0.080679
		Min	0.5767626			Min	0.470558
		Max	0.864054			Max	0.845718
	M_IBEA	Avg	**0.737086**		M_IBEA	Avg	**0.821177**
		Std D	0.095299			Std D	0.048124
		Min	0.389299			Min	0.678415
		Max	0.864054			Max	0.885620
sq3	IBEA	Avg	0.766270	sq7	IBEA	Avg	0.801778
		Std D	0.133301			Std D	0.067111
		Min	0.198802			Min	0.553095
		Max	0.7662704			Max	0.873693
	M_IBEA	Avg	**0.665162**		M_IBEA	Avg	**0.810351**
		Std D	0.140074			Std D	0.054576
		Min	0.198802			Min	0.676720
		Max	0.766270			Max	0.874023
sq4	IBEA	Avg	0.804786	sq8	IBEA	Avg	0.745065
		Std D	0.099029			Std D	0.036454
		Min	0.476124			Min	0.643070
		Max	0.876228			Max	0.814004
	M_IBEA	Avg	**0.841186**		M_IBEA	Avg	**0.811439**
		Std D	0.059579			Std D	0.050087
		Min	0.674636			Min	0.685335
		Max	0.877173			Max	0.899353

TABLE 1.4: My caption

inst	M_IBEA	M_NSGAII	EDA	GA	MMA	ACO	NewACO	PERM
sq1	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**
sq2	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**
sq3	**-8**	**-8**	**-8**	**-8**	**-8**	**-8**	**-8**	**-8**
sq4	-13	-13	**-14**	**-14**	**-14**	**-14**	**-14**	**-14**
sq5	**-23**	-22	**-23**	-22	-22	**-23**	**-23**	**-23**
sq6	**-21**	**-21**	**-21**	**-21**		**-21**	**-21**	**-21**
sq7	-35	-34	-35	-34		-34	**-36**	**-36**
sq8	**-42**	-39	**-42**	-37		-32	**-42**	-38

Bibliography

[1] Ugo Bastolla, Helge Frauenkron, Erwin Gerstner, Peter Grassberger, and Walter Nadler. Testing a new monte carlo algorithm for protein folding. *arXiv preprint cond-mat/9710030*, 1997.

[2] Henri Joseph Léon Baudrillart. *Manuel d'économie politique*. Guillaumin et cie, 1872.

[3] Andrea Bazzoli and Andrea GB Tettamanzi. A memetic algorithm for protein structure prediction in a 3d-lattice hp model. In *Applications of Evolutionary Computing*, pages 1–10. Springer, 2004.

[4] Carlos Cotta. Protein structure prediction using evolutionary algorithms hybridized with backtracking. In *Artificial Neural Nets Problem Solving Methods*, pages 321–328. Springer, 2003.

[5] Fábio L Custódio, Hélio JC Barbosa, and Laurent E Dardenne. Investigation of the three-dimensional lattice hp protein folding model using a genetic algorithm. *Genetics and Molecular Biology*, 27(4):611–615, 2004.

[6] Fábio Lima Custódio, Helio JC Barbosa, and Laurent Emmanuel Dardenne. A multiple minima genetic algorithm for protein structure prediction. *Applied Soft Computing*, 15:88–99, 2014.

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

[8] Elliackin Messias do Nascimento Figueiredo, Teresa Bernarda Orientadora Ludermir, and Carmelo José Albanez Coorientador Bastos Filho. Algoritmo baseado em enxame de partículas para otimização de problemas com muitos objetivos. 2013.

[9] Carlos M Fonseca, Joshua D Knowles, Lothar Thiele, and Eckart Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 216, page 240, 2005.

[10] Paulo HR Gabriel, Vinícius V de Melo, and Alexandre CB Delbem. Algoritmos evolutivos e modelo hp para predição de estruturas de proteínas. *Revista de Controle e Automação*, 23(1):25–37, 2012.

[11] John H Holland. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. 1975.

[12] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization. In *Genetic and Evolving Systems, 2008. GEFS 2008. 3rd International Workshop on*, pages 47–52. IEEE, 2008.

[13] Kit Fun Lau and Ken A Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989.

[14] Neal Lesh, Michael Mitzenmacher, and Sue Whitesides. A complete and effective move set for simplified protein folding. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 188–195. ACM, 2003.

[15] Gang Li, Jingfa Liu, Zhaoxia Liu, and Yu Zheng. Genetic algorithm with pull moves for folding 2d model proteins. *International Journal of Digital Content Technology and its Applications*, 6(21):412, 2012.

[16] Cheng-Jian Lin and Shih-Chieh Su. Protein 3 d hp model folding simulation using a hybrid of genetic algorithm and particle swarm optimization. *International Journal of Fuzzy Systems*, 13(2):140–147, 2011.

[17] Marco Aurélio Cavalcanti Pacheco. Algoritmos genéticos: princípios e aplicações. *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, 1999.

[18] Roberto Santana, Pedro Larranaga, and José A Lozano. Protein folding in 2-dimensional lattices with estimation of distribution algorithms. In *Biological and Medical Data Analysis*, pages 388–398. Springer, 2004.

[19] Alena Shmygelska, Rosalia Aguirre-Hernandez, and Holger H Hoos. An ant colony optimization algorithm for the 2d hp protein folding problem. In *Ant Algorithms*, pages 40–52. Springer, 2002.

[20] Alena Shmygelska and Holger H Hoos. An improved ant colony optimisation algorithm for the 2d hp protein folding problem. In *Advances in Artificial Intelligence*, pages 400–417. Springer, 2003.

[21] Christiane Regina Soares Brasil, Alexandre Cláudio Botazzo Delbem, and Daniel Rodrigo Ferraz Bonetti. Investigating relevant aspects of moeas for protein structures prediction. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 705–712. ACM, 2011.

[22] Ron Unger and John Moult. Genetic algorithms for protein folding simulations. *Journal of molecular biology*, 231(1):75–81, 1993.

[23] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pages 862–876. Springer, 2007.

[24] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.

[25] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithmsa comparative case study. In *Parallel problem solving from naturePPSN V*, pages 292–301. Springer, 1998.

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