Sunil Template

Contents

Chapter 1

Multi objective approach to the Protein Structure Prediction Problem

Ricardo H. R. Lima

Federal University of Paraná

Vidal Fontoura

Federal University of Paraná

Aurora Pozo

Federal University of Paraná

Roberto Santana

University of the Basque Country

1.1 Introduction

Proteins play a fundamental task in nature, participating in many of the most important functions of living cells. Proteins guarantee the correct functioning of a large number of biological entities in nature. Their structures are made of amino-acids as the result of the so-called protein folding process in

which the initially unfolded chain of amino-acids is transformed into its final structure. Under suitable conditions, this structure is uniquely determined by its sequence [24].

The determination of the final structure of a protein in a difficult task. Although very detailed representation of protein exists and can be used to model the protein folding, these representations are computationally very costly. This is why many authors as in [6, 14, 16, 20, 28] among others, use simplified models to represent the protein structures. A well known model for this purpose is the *Hydrophobic-Hydrophilic* model (HP model), created by Lau and Dill [17]. Considering just two types of residues H and P in a regular lattice, becomes easier to represent a protein and work with it to simulate the folding process.

The manipulation of a protein structure represented in the HP model requires some attention in order to respect the given restrictions and avoid unfeasible conformations. Another issue is the difficulty in finding good measures to verify the quality of the solution. The most common measure used for the HP model is to calculate the conformation's energy. But sometimes just the energy measure is not enough, being necessary the use of a second objective to avoid treating different solutions with the same energy value as being the same, for example.

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 [6, 14, 16, 20, 28], trying to define which methods have better results.

However, many of these approaches make use only of some well-known algorithms like Genetic Algorithm, Particle Swarm Optimization and Ant Colony Optimization. This leave a window of opportunities speaking about other classic algorithms like NSGAII [32] and IBEA [31] for example. The main difference between the mentioned methods, is that NSGAII and IBEA are multi-objective algorithms, and different from a GA, have more sophisticated mechanisms to evolve the population and try to improve the solutions.

This work proposes the application and comparison of a multi-objective approach to the Protein Folding Problem, considering two objectives. The main objective is to minimize the energy calculated from the HP model, and the second objective minimize the euclidean distance between amino acids of a protein. The evolutionary algorithms NSGAII and IBEA were used, considering some different strategies for the operators and initialization. The results of the experiments are compared with other well-known techniques and discussed to evaluate its performance.

This work is organized as follows. In Section 1.2 is presented the main aspects of the Protein Folding Problem and important information for the research. Section 1.3 is about Multi-Objective optimization, showing the positive aspects that can help solving the PSP Problem. In Section 1.4 is presented the description of the proposed method, and its relation to other method presented in a previous work. The experiments are shown in Section 1.5 with information

about the proposed method compared with other well-known works. Finally the conclusion and future works are discussed in Section 1.6.

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 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 an 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 [6, 14, 16, 20, 28] 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. Detailed representation of protein exists and can be used to model the protein folding, these representations are computationally very costly. Having this in mind, Lau and Dill [17] 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.

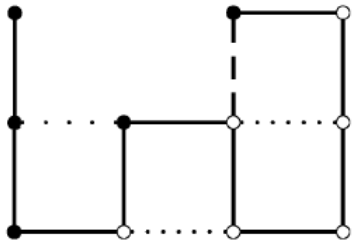

FIGURE 1.1: One possible configuration of sequence $HHHPHPPPPPH$ in the HP model. There is one HH (represented by a dotted line with wide spaces), one HP (represented by a dashed line) and two PP (represented by dotted lines) contacts.

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 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 the 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.

Among many works related to the Protein Folding Problem, here are some examples of authors and the approaches that have been used to solve it.

Lin and Su [19] proposed an mono objective EA (Evolutionary Algorithm) working with a local search strategy based on pull-moves for the PSP Problem using the simplified model Hydrophobic-Polar 2D (HP-2D). A greedy strategy was employed to avoid inconsistency in the population and to enhance the efficiency of the algorithm. The results show that this approach has better results than the GA within comparable computational times.

In other work, Lin and Su [20] 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 an improved version of a GA based on a PSO, where the solutions were based to move toward their own best solution.

Custódio and Dardenne [7] proposed a Multiple Minima Genetic Algorithm for PSP. The algorithm included a phenotype based crowding mechanism for the maintenance of useful diversity which increases the population's performance and granted the algorithm multiple solutions capabilities instead of.

Brasil *et al.*[27] proposed a multi objective algorithm in tables and compare its performance with a well-known multi-objective algorithm, the NSGAII [8],

optimizing two energy functions, both very important for the folding process: the van der Walls and Electrostatic functions.

The author of [12] also proposes the use of a multi objective algorithm in tables, similar to the method proposed by [27], however, using the HP model for the representation and solution evaluation. The author also proposes the use of a second objective that is to measure the distance between hydrophobic amino-acids, allowing the algorithm to distinguish between different solutions with the same energy value.

Unger and Moult [28] described a genetic algorithm (GA) that uses heuristic-based operators for crossover and mutation for the HP model. The algorithm outperformed many variants of Monte Carlo methods for different instances. Although the good results, the GA was unable to find the optimal solution for the longest instances considered.

The multimeme algorithm (MMA) proposed by [16] is a GA combined with a set of local search methods. The algorithm for each different instance or individuals in the population, selects the local search method that best fits. Originally used to find solutions for the functional model protein. The algorithm was later improved with fuzy-logic-based local searches, leading the algorithm to produce improved results in the PSP problem.

In [14], the author makes use of a Chain growth algorithm, called pruned-enriched Rosenbluth method (PERM), that is based on growing the sequence conformation by adding individuals particles aiming to increase good configurations and eliminating bad ones.

The ant colony optimization (ACO) [25, 26] is an algorithm that incorporates the use of a modeling step. In this approach, the artificial ants build conformations for a given HP protein sequence, apply a local search to improve the results and maintain a probability value based on the quality of the found solutions, the so called pheromone trail.

What this work proposes is the use of other well-known algorithms for multi-objective problems, considering the energy minimization and minimization of the size of a conformation as objectives, to analyze and compare its results to the values found in other works considered the state of the art.

The work of [23] describes the use of Estimation of distribution algorithms (EDAs) as an efficient evolutionary algorithm that can learn and exploit the search space regularities in the form of probabilistic dependencies. In the paper was developed new ideas about the application of EDAs to 2D and 3D simplified protein folding problems. What was analyzed is the relation between this proposal and other population-based approaches for the protein folding problem. The obtained results shows tha EDAs can obtain superior results compared with other well-known population based optimization algorithms.

1.3 Multi-objective Optimization

An Evolutionary Algorithm (EA) is an 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 Charles Darwin, the principle of natural selection favors individuals with high fitness, therefore, they have high probability of reproduction. Individuals with more descendants have more chance to perpetuate their genetic code in future generations. The genetic code is what gives the identity of each individual and is represented in the chromosomes. These principles are used in the construction of computational algorithms, that search for better solutions given a specific problem by the evolution of a population of solutions encoded in artificial chromosomes – data structures used to represent a feasible solution for a given problem in the algorithm execution [22].

Real world problems commonly have multiple objectives to minimize/maximize and are present in many areas of expertise. To optimize multi objective problems, two or more objectives are considered which are usually conflicting. For these problems it is impossible to find one unique solution. A set of solutions is reached evaluating the Pareto dominance relation [2] between the solutions. The main goal 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 real Pareto Front is an NP-hard problem [11], this way, the objective is to find a good approximation to this front.

Multi-Objective Evolutionary Algorithms (MOEAS) are extensions of EAs for multi objective problems that apply the concepts of Pareto dominance to create different strategies to evolve and diversify the solutions. In this work two MOEAs were used: NSGAII [8] and IBEA [31].

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.

The Algorithm 1 receives as inputs a parameter N for the population size and T as max number of evaluations. It starts by creating a population of size N called P_0. Then P_0 is classified according to its calculated fitness and the Non-Dominated-Sort mechanism. The classified P_0 is then submitted to a binary tournament operator to select the solutions called parents that will be used to generate new ones. The parent solutions pass through the crossover and mutation operators generating new solutions called children. At the end

of this process the children solutions are evaluated and put in a population called Q_0.

After this first step P_0 and Q_0 are put together and called as an auxiliary population R. Through the Non-dominated-sort, R is classified creating the *fronts*, where solutions from the first *front* are non-dominated by any other solution, and solutions from the second front are dominated only by the solutions of the first front, and so on. For each *front* its individuals are evaluated by the Crowding-Distance mechanism and those with higher values are stored in the next-generation population called P_t where t is the current evaluation.

After creating and filling P_t with the non-dominated solutions from all *fronts* the whole P_t has its fitness calculated and then pass through a new process of Binary Tournament, Crossover and Mutation, starting a new cycle in the algorithm.

At the end, after the stop criterion is reached, the algorithm returns a set of non-dominated solutions.

1.3.2 IBEA (Indicator-Based Evolutionary Algorithm)

Although single-objective optimization problems may have a unique optimal solution, Multi-objective problems (as a rule) offer a possibly uncountable set of solutions, which when evaluated produce vectors whose components represents trade-offs in decision space [29].

In the multi-objective context, optimizing consists in finding a front with good approximation to the *True Pareto Front*. However, is difficult to define what is the *True* Pareto Front. This way, indicators have been used to evaluate the quality of an approximation front. The *hypervolume* is an example of indicator to the evaluation and comparison of fronts.

The IBEA is an algorithm that bases its optimization process by the use of quality indicators. This means that the indicator is the way used to evaluate the non-dominated set of solutions [10].

To use the IBEA algorithm, it is necessary to 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 evaluating the convergence and diversity at the same time of the search process [15].

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

The IBEA fitness equation is given by Eq. 1.1 and is used to calculate the contribution of a given solution to the indicator value of a population, where k is a scaling factor depending on I_{Hy}, that is the quality indicator, and the underlying problem, being greater than 0, its 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 [10], based on the value of I_{Hy}, in this case, the *hypervolume*.

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.
```

The Algorithm 2 receives as parameters the population size N, max evaluations T and scale factor k. It starts by creating a population P of size N. Then it repeats the following process until reach the stop criterion: through a Binary Tournament the parents are selected to be used in the Crossover and Mutation operators to generate the children solutions and add them to a auxiliary population $\overline{P}$. After the reproduction step $\overline{P}$ is added to P. While the size of P exceeds N, the worst individual evaluated by the selected indicator is removed from the population, then the population fitness is recalculated. When the algorithm stop, it will return a set of non-dominated solutions found.

[1] *Hypervolume*: Proposed quality indicators used in the study of [32], denoted as the "size of the covered search space". This indicator has two important advantages in relation to others [30]: 1 - Sensitive to any kind of improvement in the approximation set in relation

Algorithm 2 IBEA

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

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 in 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 in front of the previous one and 2 indicates that the next residue should be placed on left from the previous. Figure 1.2 shows an example of a hypothetical chromosome and the path generated by it in the 2D lattice. The first and second aminoacids were fixed in positions (2,3) and (3,3) respectively. The third aminoacid was placed in (4,3), because the first chromosome gene is 1, and indicates that the aminoacid should be placed in front of the previous. The second chromosome gene is 0, which indicates that the next aminoacid should be placed on the right (4,4) of the previous. The fifth aminoacid was placed in (5,4) because the chromosome gene is 2 and indicates to place the next residue on the right

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.

of the previous. The de-codification of the chromosome continues until all aminoacids are placed. Note that chromosome size is always the chain length - 2 because the two first aminoacids are fixed.

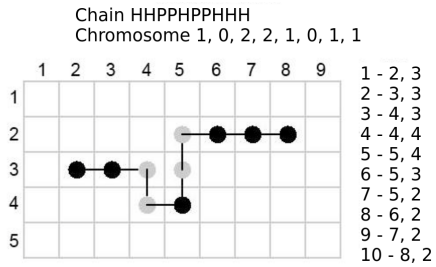

FIGURE 1.2: Example of a conformation generated by a chromosome with relative representation.

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 the single point crossover and the bit flip mutation because this combination of operators presented better results in previous experiments 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**: 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 improve the MOEAs, a pool of operators was designed based on the literature. For every mating the crossover and mutation operators are selected randomly from the pool of operators and then 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 offspring [13].
  - Two Point 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 [13].

– 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 $c = int(n \times 0.1)$ [6]. The MPX operator is usually used to promote structural diversity by performing a random shuffle between individuals, although not as thorough as a 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 [13].

– Local 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 an example of application of this operator.

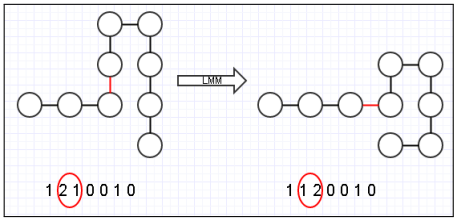

FIGURE 1.3: Example of application of the LMM operator. The genes from the red circle of left figure were swapped resulting in the right figure.

– Loop Move Mutation (LOMM): This operator is similar to LMM however this exchanges directions between genes that are five positions apart on the sequence creating a loop movement. Both LMM and LOMM are useful to generate modifications on compact structures [7].

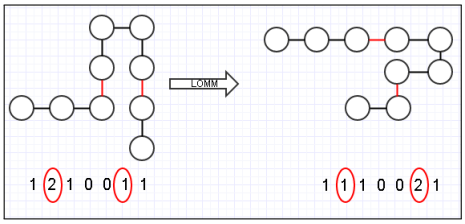

FIGURE 1.4: Example of application of the LOMM operator. The genes from the red circle of left figure were swapped resulting in the right figure.

– 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 [7]. Figure 1.4 demonstrates an application of this operator.

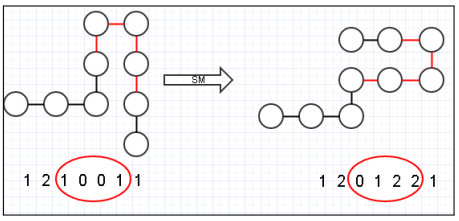

FIGURE 1.5: Example of application of the SM operator. The genes from the red circle of left figure were swapped by random genes resulting in the right figure.

– 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 residues can be mapped to its inverse. Figure 1.4 presents an example of application of this operator.

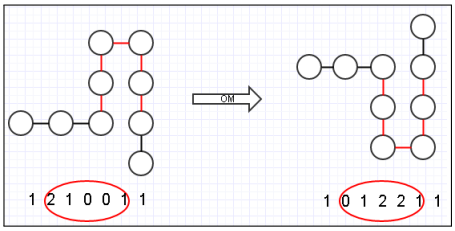

FIGURE 1.6: Example of application of the OM operator. The genes from the red circle of left figure were swapped by random genes resulting in the right figure.

- **Backtrack Initialization**: Traditionally the initial population of solutions are generated randomly in the presented MOEAs. The random based generation of the solutions has 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 avoid this problem a backtrack strategy should be applied. In this approach 20 percent of the initial population was generated using the backtrack initialization.

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

- **Energy value**: This is the main objective and consists in the energy of given protein conformation. The goal is to minimize the energy value and it is calculated as described in section 1.2.1. This objective guides the search progress towards regions that the energy associated to the protein conformations are minimal, thus, achieving protein conformations which are closest to native structure of a protein.
- **Minimize the distance between the two farthest residues**: This is a secondary objective and it was inspired by the related work [12]. 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 allows the generation of invalid solutions using the HP-2D model. A solution is considered invalid when it solution does not perform a self-avoiding walk (SAW) as mentioned before. In other words, an invalid solution is when a given residue collides with another already placed on the lattice. A simple mechanism for repairing these situations was developed, and the code can be seen in the Algorithm 3.

Algorithm 3 Mechanism to repair infeasible solutions

1. The direction of the next residue should be placed is obtained 2. Verification if the direction selected will cause collision.
3. If a collision is detected, a new direction is used.
4. Repeat the steps 2 and 3, until is possible to place the next residue, or if all directions were tested and cause collisions.
5. If it 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.

This mechanism was implemented because in previous experiments it was observed that the number of infeasible solutions was very high. It is necessary to mention that even with the mechanism to repair solutions, there are still infeasible solutions because the mechanism cannot always repair them. Thus, infeasible solutions are penalized by adding the number of collisions to the energy value. This mechanism is used by the evaluation process of both approaches described before (MOEAs without any modifications and the MOEAs supported by the backtracking initialization and the pool of operators).

To evaluate and compare the performance of multi-objective algorithms, quality indicators are commonly used. In this study it was used the hypervolume indicator, which considers the volume of the search space dominated by

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	$H^{12}PHPH\{PPHH\}^2PPH\{PPHH\}^2$ $PPH\{PPHH\}^2PPHPHPH^{12}$

the known Pareto front [33] of an algorithm. Higher hypervolume value means that the quality of an algorithm is better than one with lower hypervolume value.

Both approaches were implemented using the open source architecture from jMetal framework [9]. jMetal is easy to extend, has a well-organized structure and also an active community.

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 previous works such as [1, 25, 28, 5, 24, 26, 18]. 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. It is worthwhile to mention that the sequences used in this chapter were randomly generated. Hence they do not fold to a single conformation, as natural proteins, because they are not products of natural selection [4].

The configuration used for the MOEAs was defined based on the sequence length. For smaller sequences it was used a smaller population size and for larger sequences it was used a larger population size. The same is true in the case of the stop condition (max evaluations). Table 1.2 presents the population size and maximum number of evaluations used for each amino-acid sequence.

TABLE 1.2: Population size and maximum number of evaluations configurations for each sequence

Sequences	Size	Population Size	Max Evaluations
sq1	20	100	25000
sq2	24	100	25000
sq3	25	500	250000
sq4	36	500	250000
sq5	48	1000	2500000
sq6	50	1000	2500000
sq7	60	2500	2500000
sq8	64	2500	2500000

In the case of the first approach the probability of crossover/mutation occurrence was fixed, for all sequences, in 0.9 and 0.01 respective. The second approach does not uses a probability since the operators are always applied to generate new individuals. The auxiliary population size used by the IBEA algorithm was fixed in 200 for all sequences.

1.5.1 Comparison between the modified/traditional versions of the MOEAs

As mentioned in the end of section 1.4 the hypervolume indicator was used in order to compare the MOEAs performance. The hypervolume results are presented on table 1.3. The hypervolume average and standard deviation, of 30 independent executions, are presented. The average values highlighted with a bold font are the highest values. Looking to table 1.3 is possible to notice, except for $sq1$, that for all sequences the M_IBEA (modified version of the IBEA with backtrack and pool of operators) obtained a higher hypervolume average than the other algorithms. In the case of $sq1$ the IBEA without modifications obtained a higher value compared with the others. It is also possible to see, comparing only the NSGAII versions, that the modified version M_NSGAII obtained better results, except for $sq1$. In general, the MOEAS with backtrack and pool of operators presented an improvement in relation to the traditional MOEAs. The cells from M_IBEA that are marked with gray presented statistical difference according to the Kruskal-Wallis test [21] between M_IBEA and all other algorithms (NSGAII, M_NSGAII and IBEA).

1.5.2 Comparison with previous works

This section presents the comparison of the results obtained by the MOEAs with other approaches from the previous works described on section 1.2.1, and is only concerned with the first objective. (Energy of given conformation),

TABLE 1.3: Results of hypervolume average/standard deviation of the MOEAs

Instance	Hypervolume Average (Std D)			
	NSGAII	M_NSGAII	IBEA	M_IBEA
sq1	0.742827 (0.106315)	0.720864 (0.131351)	**0.789712** (0.067660)	0.786571 (0.099424)
sq2	0.680572 (0.083445)	0.712275 (0.137226)	0.719960 (0.080727)	**0.737086** (0.095299)
sq3	0.671171 (0.129417)	0.709898 (0.124201)	0.716438 (0.148112)	**0.738017** (0.155638)
sq4	0.702280 (0689832)	0.740153 (0.075271)	0.751755 (0.092427)	**0.785728** (0.055607)
sq5	0.707654 (0.082611)	0.758128 (0.062315)	0.733464 (0.128757)	**0.807637** (0.039620)
sq6	0.667771 (0.132218)	0.774017 (0.063231)	0.728699 (0.080679)	**0.821177** (0.048124)
sq7	0.784483 (0.063257)	0.792843 (0.033062)	0.801778 (0.067111)	**0.810351** (0.054576)
sq8	0.677464 (0.041287)	0.705798 (0.053048)	0.7450656 (0.036454)	**0.811439** (0.050087)

since the other works are single-objective. Table 1.4 presents the best results, in terms of energy, found by the modified MOEAs and also the best results obtained by the previous works.

For the first 3 sequences $sq1$, $sq2$ and $sq3$ the modified MOEAs (M_NSGAII and M_IBEA) obtained the same results that the previous works obtained. In the case of $sq4$ both M_IBEA and M_NSGAII obtained a value of -13 and all the previous works obtained the optimum value of -14. For sequence $sq5$ four previous works and M_IBEA have achieved the optimum value -23. However M_NSGAII and the other previous works obtained a lesser value of -22. In the case of sequence $sq6$ all algorithms obtained the optimum value of -21. For sequence $sq7$ the M_IBEA obtained -35 as the EDA [23]. However the best value found for $sq7$, -36, were obtained by NewACO [26] and PERM [14]. For the last sequence $sq8$ the M_IBEA obtained the optimum value of -42 which is the same obtained by EDA [23] and NewACO [26]. All other approaches obtained lesser values for sequence $sq8$.

TABLE 1.4: Comparison with the previous works

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

1.6 Conclusion and Future works

MOEAs are evolutionary algorithms to address the challenge of optimization of multiple objectives at the same time. They have been showing good results in many areas of science. At this chapter two well known MOEAs were applied in order to address the PSP problem using the HP-2D model. Two multi-objective approaches were presented: the first approach utilizes the IBEA and NSGAII algorithms as is, without any modification; the second approach consisted on modifying IBEA and NSGAII, adding backtrack initialization and a pool of operators, in order to enhance the results. Given the experiments results it became clear that the second approach was able to explore better the search space than the first approach. Also it was possible to check that the multi-objective approach using the IBEA algorithm, modified with backtrack and a pool of operators, obtained competitive results when compared with the previous works. However the NSGAII and M_NSGAII algorithms did not present satisfactory results.

Future works include to explore better selection methods to select the operators from the pool operators and also the addition of more operators. It is believed that the pool of operators is the most responsible for the improvement in the exploration of the MOEAs. Therefore more intelligent selection mechanism, which considers the history of the operators application, could improve even more the performance of the MOEAs. The extension of this work to the HP-3D is also planned for the future works. The HP-3D model is more complex than the HP-2D it is possible that the multi-objective approach, presented in this chapter, could suit well. The application of hyper-heuristics is also planned for generation of specialized optimization algorithms for the PSP problem.

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] HS Chan and E Bornberg-Bauer. Perspectives on protein evolution from simple exact models. *Applied bioinformatics*, 1(3):121–144, 2001.

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

[6] 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.

[7] 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.

[8] 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.

[9] Juan J Durillo and Antonio J Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[10] 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.

[11] 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.

[12] 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.

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

[14] Hsiao-Ping Hsu, Vishal Mehra, Walter Nadler, and Peter Grassberger. Growth algorithms for lattice heteropolymers at low temperatures. *The Journal of chemical physics*, 118(1):444–451, 2003.

[15] 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.

[16] Natalio Krasnogor, BP Blackburne, Edmund K Burke, and Jonathan D Hirst. Multimeme algorithms for protein structure prediction. In *Parallel Problem Solving from NaturePPSN VII*, pages 769–778. Springer, 2002.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] P. E. McKight and J. Najab. Kruskal-Wallis test. *Corsini Encyclopedia of Psychology*, 2010.

[22] 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.

[23] Roberto Santana, Pedro Larrañaga, Jose Lozano, et al. Protein folding in simplified models with estimation of distribution algorithms. *Evolutionary Computation, IEEE Transactions on*, 12(4):418–438, 2008.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] David A Van Veldhuizen and Gary B Lamont. Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, pages 221–228. Citeseer, 1998.

[30] 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.

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

[32] 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.

[33] 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.