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 folding

We will briefly recall some of the main biological concepts related to the protein folding problem that are relevant to our discussion.

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.

The achievement of the protein native structure is the result of the so-called protein folding process. The laws that govern protein folding are unknown. Therefore a number of ideas have emerged that try to answer this question: how do amino acid sequences specify proteins 3-d structure?

There are two main approaches to protein folding, commonly referred as the "classical" and "new" views. The "classical" view considers folding as a defined sequence of states leading from the unfolded to the native state. This sequence is called the pathway [9]. In the "new" view approach, folding is seen as the progressive organization of an ensemble of partially folded structures through which the protein passes on its way to the folded structure [7]. This approach emphasizes the idea of each state being an ensemble of rapidly inter-converting conformations. One of the main differences between both approaches is that the "new" view allows for a more heterogeneous transition state than the "classical" view, which concentrates on a single, well-defined folding pathway [1].

Figure 1.2 shows one schematic representation of the "classical" (left) and "new" (right) views of protein folding. In the figure, each possible protein configuration is represented as a circle, and arrows represent possible transitions between configurations. In both approaches, the native state (filled circle) is achieved when the energy is minimized.

SinglePointCrossover TwoPointsCrossover MultiPointCrossover (para crossovers)

BitFlipMutation LoopMoveOperator LocalMoveOperator SegmentMutationOperator OppossiteMoveOperator (para mutacoes)

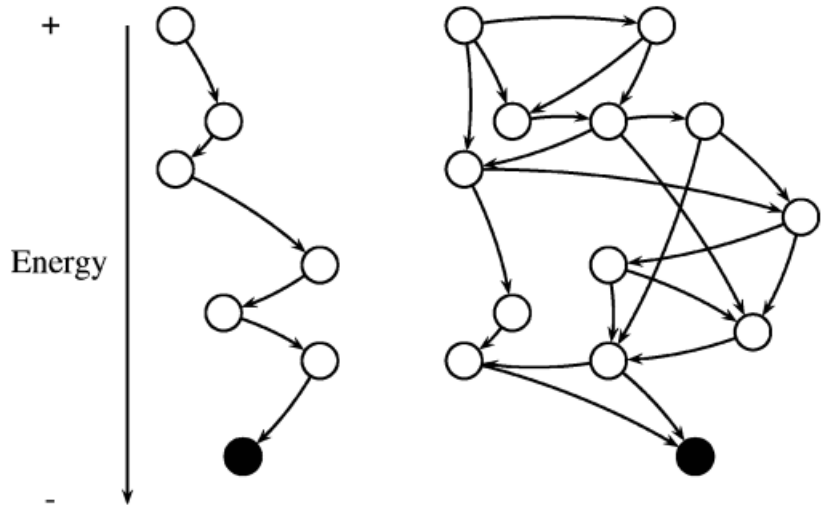

FIGURE 1.1: Schematic representation of the classical (left) and new (right) views of protein folding.

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 [8].

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 [5], 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 [3] and IBEA [11].

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).

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 [4]. 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 [6].

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

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

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.
```

loss measure of the approximation to the Pareto front if the solution x_i was removed of the population [4], 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).

[1] *Hypervolume*: Proposed quality indicators used in the study of [12], denoted as the "size of the covered search space". This indicator has two important advantages in relation to others [10]: 1 - Sensitive to any kind of improvement in the approximation set in relation 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.

Algorithm 2 IBEA

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

1.4 Proposed method

The proposed method consists in the application of the algorithms NSGAII and IBEA to the PSP problem using the relative representation applied to the HP model. The multi-objective optimization framework jMetal [**?**] was used, it presents implementation of the used MOEAs and a easy way to personalize them. To evaluate the generated solutions, a evaluation mechanism was implemented considering two objectives:

1. Maximize the number of topological neighbors HH (main objective);
2. Minimize the maximum euclidean distance between residues (secondary objective).

The first objective guide the search aiming find a solution that generates a structure where the energy value is minimum (maximizing the number of topological neighbors HH), this way, obtaining a structure closer of the native conformation of the protein. The second objective allows to differ solutions with the same value of energy but with different compression degrees. The more compact was the max value of euclidean distance, more compact will be the generated conformation.

The chromosomes are represented by integer vectors where the genes specify which direction, related to the previous residue, the next one should be placed. The genes can assume one of three values 0, 1 and 2, where 0 means

the next residue should be placed at right of the previous one, 1 means the next residue should be placed in front of the previous and 2 means the next residue should be placed at left of the previous one. The Figure 1.2 demonstrate a example of a possible chromosome to a chain of 10 residues and its generated conformation.

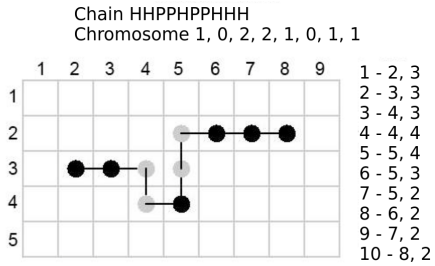

FIGURE 1.2: Example of a conformation generated by a chromosome with relative representation.

The relative representation it is subject to the generation of infeasible solutions using the HP model. A solution is considered infeasible when a 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.

Algorithm 3 Mechanism to repair infeasible solutions

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

This mechanism was implemented because in early experiments was observed that the number of infeasible solutions was too big. It is necessary mention that the even with the mechanism to repair solutions, there are still infeasible solutions because the mechanism can't always repair. Yet, infeasible solutions are penalized by subtracting the number of collisions to the quantity of topological neighbors.

To evaluate and compare the performance of the 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 true front [**?**]. The high the hypervolume is, better the quality of the front found by one of the algorithms.

1.5 Experiments

1.6 Conclusion

Bibliography

[1] David Baker. A surprising simplicity to protein folding. *Nature*, 405:39–42, 2000.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] J. N. Onuchic and P. G. Wolynes. Theory of protein folding. *Current Opinion in Structural Biology*, 14:70–75, 2004.

[8] 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.

[9] V. S. Pande, A. Y. Grosberg, T. Tanaka, and D. S. Rokhsar. Protein folding pathways: Is a 'new view' needed? *Current Opinion in Structural Biology*, 8(1):68–79, 1998.

[10] 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.

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

[12] 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.