Protein Structure Prediction Using Evolutionary Algorithms Hybridized with Backtracking

Carlos Cotta

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
University of Málaga, Campus de Teatinos, 29071 - Málaga - SPAIN
ccottap@lcc.uma.es

Abstract. This work explores different evolutionary approaches to Protein Structure Prediction (PSP), a highly constrained problem. These are the utilization of a repair procedure, and the use of evolutionary operators whose functioning is closed in feasible space. Both approaches rely on hybridizing the evolutionary algorithm (EA) with a backtracking algorithm. The so-obtained hybrid EAs are described, and empirically compared to a penalty-based EA. The utilization of the repair procedure reveals itself as a very appropriate technique for tackling this problem.

1 Introduction

Proteins are biomolecules of paramount importance for life as we know it: they play an essential role in many biological functions, acting as carriers, catalysts, regulators, etc. In essence, a protein is a sequence of aminoacids. When left in the appropriate environmental conditions, this sequence exhibits the extraordinary property of folding itself, quickly reaching a unique low-energy state. Such state is termed *native state*, and it ultimately determines the biological function of the protein. The extreme importance of being capable of ascertaining the native conformation of a protein from its amino-acid sequence is thus clear (for example, it is useful for designing new drugs for a target disease). This is known as the Protein Structure Prediction (PSP) problem.

It turns out that solving PSP instances to optimality is a very hard problem, even when simplified models are considered [1]. For this reason, the use of heuristic techniques such as Evolutionary Algorithms (EAs) is in order. EAs have been applied to the PSP problem in a number of works, e.g., [4, 5, 8], with moderate success. One of the difficulties that such an application has to deal with is the existence of geometrical constraints in the final conformation of the protein (i.e., self-avoidance in the chain, forbidden torsion-angles, etc.). This difficulty has been usually tackled using a penalty function that measures to which extent these constraints are violated. Thus, infeasible solutions are allowed, but they are assigned a lower fitness value due to the existence of a penalizing term (e.g., see for example [4, 5]).

This work explores alternatives to the penalty approach mentioned above. More precisely, we consider the utilization of a repair procedure (mapping infeasible solutions to feasible conformations), and the use of evolutionary operators whose functioning is closed in feasible space. Both approaches rely on a backtracking algorithm, tailored to the PSP problem. The combination of this backtracking algorithm with the EA results in a *hybrid* algorithm. The so-obtained hybrid EAs will be described, and empirically compared to a penalty-based EA.

The remainder of this paper is organized as follows. First, Section 2 provides the necessary background on the PSP models considered in this work. Then, Section 3 describes the application of EAs to the PSP, focusing on the backtracking algorithm located at the core of the hybrid, as well as on the evolutionary operators based on this algorithm. Subsequently, experimental results for the different EAs considered are reported in Section 4. Finally, Section 5 presents some conclusions are outlines future work.

2 A Gentle Introduction to Protein Structure Prediction

As mentioned in the previous section, a protein is a sequence of aminoacids. Each of these aminoacids can be from one out of twenty different types, and it is connected to its neighbors in the sequence by a *peptide* bond. While this bond is relatively rigid, a certain amount of rotation can take place around other atomic links. Such rotation is responsible for the folding of the protein. A realistic simulation of the folding process should then take into account the physical and chemical factors affecting such rotations. This is time consuming and computationally expensive, and hence simplified models are needed.

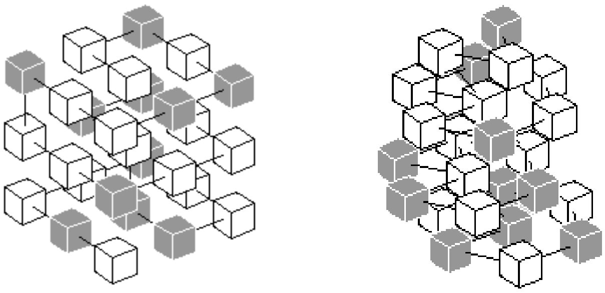

Fig. 1. Examples of protein conformations in the cubic lattice (left) and in the cube-octahedral lattice (right) under the HP model. Dark (resp. white) boxes represent hydrophobic (resp. hydrophilic) aminoacids.

One of the most popular of such models is the *Hydrophobic-Hydrophilic* model (HP model) of Dill [2]. In this model, each aminoacid is classified into two classes: hydrophobic or non-polar (H), and hydrophilic or polar (P), according to their interaction with water molecules. In addition, the sequence is assumed to be embedded in a certain lattice. This lattice is used to discretize the space of conformations, and can exhibit different topologies. The simplest one is the square lattice, shown in Fig. 1 (left) for three dimensions. Other typical examples are tetrahedral [3] and cube-octahedral [9] lattices (shown in Fig. 1–right).

Any feasible conformation in the HP model is assigned a free energy level. To be precise, each pair of hydrophobic aminoacids being topological neighbors in the conformation contributes a contact free energy $\epsilon < 0$, provided that these two aminoacids are not adjacent in the sequence; any other topological contact does not contribute anything to the total free energy. Notice that the native state of a protein is a low-energy conformation (it is actually conjectured to be the global minimum). Thus, the number of `HH` contacts is maximized in the native state.

As shown in [1], finding this globally optimal conformation under the HP model is NP-hard. This justifies the utilization of heuristic techniques such as EAs for solving this problem. Next section will describe the EA deployment on the PSP problem.

3 Evolutionary Approaches to the PSP Problem

The application of EAs to the PSP problem involves determining appropriate representation and operators, as well as defining a suitable fitness function. We will start by briefly discussing these aspects. This will pave the way for introducing hybrid operators that try to overcome the limitations of classical evolutionary approaches.

3.1 Basic Setting

According to the description of the HP model provided in Section 2, a protein conformation is an embedding of the corresponding sequence in a certain lattice. Each individual in the EA must thus represent such an embedding. This is typically done by using internal coordinates, i.e., the folding is expressed as a sequence of *moves* specifying the location of each aminoacid with respect to the previous one (the location of the first aminoacid in the sequence is fixed, and hence $n-1$ moves must be given in order to specify a folding for a sequence of n aminoacids). Obviously this representation depends on the particular lattice topology considered; for example, each location has 6 neighbors in a cubic lattice, and 12 neighbors in a cube-octahedral lattice. This raises a second issue, i.e., the precise representation of each move.

Two major schemes for representing internal moves can be found in the literature. First of all, we can consider the *absolute* representation [10]. In this representation, an absolute reference system is assumed, and moves are specified with respect to it. As an example, consider the case of the cubic lattice; there are 6 possible absolute moves from a given location: North, South, East, West, Up and Down (see Fig. 2–left). Thus, a conformation is expressed as a sequence $s \in \{\texttt{N, S, E, W, U, D}\}^{n-1}$, where n is the length of the protein sequence. As an alternative, *relative* moves [7] can be considered. In this case, the reference system is not fixed, but it depends on the last move. This is illustrated in Fig. 2–right; as it can be seen, five moves are allowed: Forward, Turn Up, Turn Down, Turn Right, and Turn Left. Hence, conformations are expressed as sequences $s \in \{\texttt{F}, \texttt{T}_\texttt{U}, \texttt{T}_\texttt{D}, \texttt{T}_\texttt{L}, \texttt{T}_\texttt{R}\}^{n-1}$.

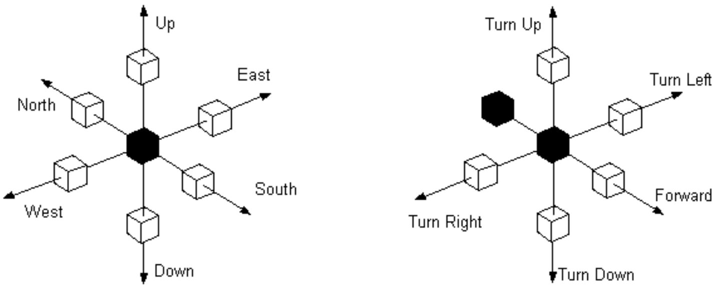

Fig. 2. (Left) Absolute moves in a cubic lattice. The black cube represents the current location. (Right) Relative moves in a cubic lattice. The black cubes represent the current location and the previous one.

It is clear that many sequences of moves do not correspond to feasible conformations, since the self-avoidance constraint will be violated. This is especially relevant in EAs due to the fact that standard reproductive operators will likely produce infeasible offspring even when the parents were feasible. As previously anticipated, the classical approach for dealing with this situation is allowing such infeasible solutions, but penalizing them at the evaluation stage. More precisely, let $D = \{d_{ij}\}$ be a matrix such that d_{ij} is the distance between p_i and p_j, respectively the ith and the jth aminoacids in the protein[1]. Then, the objective function (to be minimized) has the following structure:

$$f(D) = \sum_{i=1}^{n-2} \left[O(D,i) \sum_{j=i+2}^{n} O(D,j) E(p_i,p_j) \delta(d_{ij},1) \right] + C \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta(d_{ij},0) \quad (1)$$

where $\delta(\cdot,\cdot)$ is the Kronecker-delta function, $O : \mathbb{N}^{n\times n} \times \mathbb{N} \to \{0,1\}$ is

$$O(D,i) = \prod_{j \neq i} [1 - \delta(d_{ij},0)] \ , \qquad (2)$$

i.e., $O(D,i) = 0$ if, and only if, the move sequence produces an overlap involving the ith aminoacid, $E : \{H,P\}^2 \to \mathbb{Z}$ is a function capturing the free contact energy between two certain aminoacids[2], and $C > 0$ is a constant that weights the penalty infringed to infeasible solutions.

By using the evaluation function shown above, the search space can be explored as if no constraint existed, i.e., standard operators can be used for recombination (e.g., single-point crossover) and mutation (e.g., random gene-substitution). Additionally, *ad hoc* operators can be defined in orden to exploit

[1] The distance is measured as the length of the shortest path in the lattice connecting their locations after applying the folding sequence.

[2] In this case, $E(\mathtt{H},\mathtt{H}) = -1$, being zero otherwise.

some features of the problem. A typical example is *specular mutation* (see [6] for example), an operation that flips a part of the folded sequence along a certain symmetry axis, e.g., by changing `N` by `S` and *vice versa*.

3.2 The Backtracking Algorithm

As mentioned in Section 1, the PSP problem is probably intractable, and hence the use of exact techniques such a Branch-and-Bound (BnB) or Backtracking is inherently limited by a complexity barrier. Of course, this limitation refers to the deployment of these exact techniques for finding the globally optimal solution. However, the utilization of these techniques for finding feasible solutions is in principle perfectly affordable.

This section describes a backtracking algorithm aimed at producing feasible solutions for a certain PSP problem instance. This particular algorithm constitutes a simple yet efficient approach for the purposed task. Its pseudocode is shown in Figure 3. The algorithm receives three parameters. The first one is τ, a table containing the allowed moves for each aminoacid in the protein (for each one but the first, to be precise); thus, τ_k is a list of allowed moves for the $(k+1)$-th aminoacid and $\tau_{k,r}$ is the rth move. Although τ may contain in principle the full set of moves, in general $|\tau_k|$ will not be the same for every k. This will be illustrated in the following subsection.

The second parameter is s, a partial conformation involving $|s|$ aminoacids. As to the third parameter, it is a Boolean flag used to finalize the execution of the algorithm as soon as a feasible conformation is found. Notice finally that :: represents the sequence concatenation operator. Next subsection will be devoted to describe the hybridization of the EA with this basic backtracking algorithm.

```
PSP-Backtracking ($\downarrow$ $\tau$:move[][], $\downarrow\uparrow$ $s$:move[], $\uparrow$ $solutionFound$:bool)

    if $feasible(s)$ then
        if $|s| = n - 1$ then
            $solutionFound \leftarrow$ TRUE
        else
            $solutionFound \leftarrow$ FALSE
            $i \leftarrow 1$
            while $\neg solutionFound \wedge (i \leq |\tau_{|s|}|)$ do
                $s' \leftarrow s :: \langle \tau_{|s|,i} \rangle$
                PSP-Backtracking ($\tau$, $s'$, $solutionFound$)
                $i \leftarrow i + 1$
            endwhile
            if $solutionFound$ then
                $s \leftarrow s'$
            endif
    else
        $solutionFound \leftarrow$ FALSE
    endif
```

Fig. 3. Pseudocode of a Backtracking algorithm for finding feasible conformations.

3.3 Backtracking-based Evolutionary Operators

Besides the penalty approach sketched in Subsection 3.1, the PSP problem can be dealt using a repairing procedure, or feasible-space operators. Both approaches can be implemented via the use of the backtracking algorithm presented above.

Let us start by considering the feasible-space approach. This involves the EA having a population of feasible solutions at all times. First of all, this implies that the initial population must be composed of such feasible solutions. To do so, it suffices to use the backtracking algorithm using a table τ such that τ_k is a different random permutation of all moves. This will produce a random feasible conformation each time the backtracking algorithm is invoked.

As to recombination and mutation, they must respect feasibility of the solutions they produce. Focusing firstly on recombination, let $\eta = \langle \eta_1, \cdots, \eta_{n-1} \rangle$ and $\zeta = \langle \zeta_1, \cdots, \zeta_{n-1} \rangle$ be two feasible conformations; they can be recombined in feasible space by using the backtracking algorithm with τ_k being $\langle \eta_k \rangle$ if $\eta_k = \zeta_k$, and a random permutation of $\langle \eta_k, \zeta_k \rangle$ otherwise. This will provide a random feasible combination of the parental information without introducing exogenous information (each move in the descendant will be taken from one of the parents). Finally, mutation is performed by selecting an aminoacid i in the individual η, and assigning a random move ξ $(\neq \eta_i)$ to it. Subsequently, the backtracking algorithm is invoked having τ_k $(k \neq i)$ being a permutation of all moves such that $\tau_{k,1} = \eta_k$, and $\tau_i = \langle \xi \rangle$. This will produce a feasible solution with the mutated move, and that will have the original values in the remaining moves except where a change be required to avoid a superposition.

Notice finally that the repair-based approach can be implemented using the mutation operator described above as the repairing mechanism (it will produce a feasible solution no matter the feasibility/infeasibility of the solution to be mutated).

4 Empirical Results

The experiments have been done with an elitist generational EA ($popsize = 100$, $p_c = .9$, $p_m = 0.01$) using linear ranking selection ($\eta = 2.0$). A maximum number of 10^5 evaluations has been enforced. In order to provide a fair comparison, the internal backtracking steps performed by some operators have been accounted and deducted from this computational limit.

The problem instances considered are taken from [10], and are labeled as UMxx, where xx is the number of aminoacids in the sequence. Several lattice models have been used, including cubic and cube-octahedral topologies. Due to space limitations we focus here on the results obtained on the three-dimensional cubic lattice. These are shown in Table 1 for the three EA approaches. The reproductive operators used have been SPX (P-EA and R-EA), backtracking recombination (F-EA), random gene-substitution[3] (P-EA), and backtracking mutation (F-EA and R-EA). In all cases, initialization is done using only feasible solutions.

[3] Specular mutation has been tried as well, with worse results.

Table 1. Results of the different EA approaches (averaged for 50 runs).

Penalty-based Approach (P-EA)						
	Absolute Encoding			Relative Encoding		
sequence	best	mean ± σ	median	best	mean ± σ	median
UM20	11	10.32 ± 0.71	10.5	11	9.02 ± 0.95	9
UM24	13	10.84 ± 1.01	11	11	8.60 ± 1.00	8.5
UM25	9	8.00 ± 0.82	8.5	9	6.78 ± 1.04	7
UM36	18	14.70 ± 1.24	14	15	11.36 ± 1.60	12.5
UM48	26	22.10 ± 1.73	23	22	16.50 ± 2.33	19
UM50	25	20.46 ± 1.72	22	21	14.94 ± 1.87	15.5
UM60	43	36.64 ± 2.71	40	37	29.60 ± 3.14	28.5
UM64	41	36.28 ± 2.40	34.5	36	26.72 ± 3.06	24
Feasible-space Approach (F-EA)						
	Absolute Encoding			Relative Encoding		
sequence	best	mean ± σ	median	best	mean ± σ	median
UM20	11	10.32 ± 0.61	10	11	9.84 ± 0.86	10.5
UM24	13	10.90 ± 0.98	11	11	10.00 ± 0.87	9.5
UM25	9	7.98 ± 0.71	8	9	8.64 ± 0.69	8
UM36	18	14.38 ± 1.26	14.5	18	13.72 ± 1.41	15.5
UM48	25	20.80 ± 1.61	20	28	18.90 ± 2.08	22.5
UM50	23	20.20 ± 1.50	21.5	22	19.06 ± 1.46	19
UM60	39	34.18 ± 2.31	33.5	38	32.28 ± 3.09	36.5
UM64	39	33.01 ± 2.49	37.5	36	30.84 ± 2.55	30
Repair-based Approach (R-EA)						
	Absolute Encoding			Relative Encoding		
sequence	best	mean ± σ	median	best	mean ± σ	median
UM20	11	10.52 ± 0.54	10.5	11	10.26 ± 0.69	10.5
UM24	13	11.28 ± 0.90	11	13	10.36 ± 0.88	10.5
UM25	9	8.54 ± 0.64	8.5	9	8.18 ± 0.79	8.5
UM36	18	15.76 ± 1.05	16	16	14.16 ± 1.24	15.5
UM48	28	24.60 ± 1.57	26.5	26	21.28 ± 1.64	22
UM50	26	23.02 ± 1.48	23.5	24	20.06 ± 1.47	21.5
UM60	49	41.18 ± 2.75	39.5	43	36.92 ± 2.45	38
UM64	46	40.40 ± 2.50	40	40	35.10 ± 2.46	36.5

Notice first of all that the results for EAs using the absolute encoding are better than those of the EAs using relative encondings. The differences are in some cases small, and not very significant, but are appreciable in the larger instances. Notice also that the results of the F-EA are in general worse than those of the P-EA, thus providing some support to the claims made in [5] regarding the limitations of the first approach for traversing the search space. Nevertheless, notice that the R-EA provide the best results; jumping from an infeasible conformation to a nearby feasible one thus seems to be an appropriate strategy for exploring the search space in this problem.

5 Conclusions

The application of EAs to the PSP problem has been commonly approached via penalty functions (P-EA). The motivation is twofold: on one hand, it results in simpler algorithms; on the other hand, it has been claimed that handling infeasible solutions is necessary in order to efficiently traverse the search space. This work has been aimed at studying the performance of two alternative approaches: a feasible-space EA (F-EA) and a repair-based EA (R-EA). Both EAs rely on the use of a embedded backtracking algorithm.

Several conclusions can be drawn from the experiments realized. First of all, the need for handling infeasible solutions has been supported but only to some extent. Although the F-EA provides comparatively worse results than the P-EA, the fact that the population is initialized with feasible solutions plays a major role in the good performance of the latter. Furthermore, the R-EA provides good results, suggesting the usefulness of exploring feasible conformations in the neighboring regions of those infeasible solutions generated by the EA. Additionally, and from an algorithmic point of view, the resulting algorithms are not much more complex to implement than the P-EA.

Future work will be directed to generalize these results to other folding models, as well as to investigate the possibilities for adding local improvement operators that turned the EA into a full-featured memetic algorithm.

Acknowledgements This work is partially supported by Spanish MCyT and FEDER under contract TIC2002-04498-C05-02. The author wishes to thank M. Ruiz for valuable assistance.

References

1. B. Berger and T. Leight. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(2):27–40, 1998.
2. K.A. Dill. Dominat forces in protein folding. *Biochemistry*, 29:7133–7155, 1990.
3. D. Hinds and M. Levitt. A lattice model for protein structure prediction at low resolution. *Proceedings of the National Academy of Sciences*, 89:2536–2540, 1992.
4. M. Khimasia and P. Coveney. Protein structure prediction as a hard optimization problem: the genetic algorithm approach. *Molecular Simulation*, 19:205–226, 1997.
5. N. Krasnogor, W.E. Hart, J. Smith, and D.A. Pelta. Protein structure prediction with evolutionary algorithms. In W. Banzhaf et al., editors, *Proceedings of the GECCO'99*, pages 1596–1601, San Mateo CA, 1999. Morgan Kaufmann.
6. N. Krasnogor, D.A. Pelta, P.M. López, P. Mocciola, and D. de la Canal. Genetic algorithms for the protein folding problem: A critical view. In E. Alpaydin, editor, *Proceedings of the EIS'98*. ICSC Academic Press, 1998.
7. A.L. Patton, W.F. Punch III, and E.D. Goodman. A standard GA approach to native protein conformation prediction. In L.J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 574–581, San Mateo CA, 1995. Morgan Kaufmann.
8. A. Piccolboni and G. Mauri. Application of evolutionary algorithms to protein folding prediction. In Jin-Kao Hao et al., editors, *Artificial Evolution III*, volume 1363 of *Lecture Notes in Computer Science*, pages 123–136. Springer-Verlag, 1998.
9. G. Raghunathan and R.L. Jernigan. Ideal architecture of residue packing and its observation in protein structures. *Protein Science*, 6:2072–2083, 1997.
10. R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231(1):75–81, 1993.