A Memetic Algorithm for Protein Structure Prediction in a 3D-Lattice HP Model

Andrea Bazzoli and Andrea G.B. Tettamanzi

Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013 Crema (CR), Italy
andrea.tettamanzi@unimi.it

Abstract. This paper presents a memetic algorithm with self-adaptive local search, applied to protein structure prediction in an HP, cubic-lattice model. Besides describing in detail how the algorithm works, we report experimental results that justify important implementation choices, such as the introduction of speciation mechanisms and the extensive application of local search. Test runs on 48-mer chains show that the proposed algorithm has promising search capabilities.

1 Introduction

After the successful completion of the Human Genome Project, the focus of biological investigation has shifted to the study of proteins and their interactions, which are a key to understanding the metabolism of cells. Since the biological activity of a protein depends on its 3D (tertiary) structure, determining that structure has become one of the most central problems in Bioinformatics.

A number of computational methods have been developed over the years, in order to simulate the folding process of linear chains of amino-acid residues (primary sequences) to their tertiary conformations, also called native states. These methods, however, are computationally very expensive, and their applicability is limited to relatively short polymer sequences.

An alternative approach is to "predict" the 3D structure given the primary sequence, by using some suitable heuristics, which can be statistical (like comparative modeling) or optimization-based.

1.1 Problem Statement

The protein structure prediction (PSP) problem can be stated as an optimization problem, whereby the minimum of a specific energy function is sought in the space of all possible conformations. As a matter of fact, according to Anfinsen's hypothesis [1], the native state of a protein can be seen as the one with the lowest energy.

Given the considerable number of amino acids in a protein, all-atom energy functions are often prohibitive on today's computers. For this reason, researchers have developed a great variety of simplified protein models, which speed up the calculations by somewhat reducing the number of degrees of freedom (and therefore parameters) involved. Although their descriptions are inevitably less precise, these models offer nonetheless a global perspective of protein structures, which has proven helpful in confirming or questioning important theories.

One remarkable simplified energy function stems from the so-called HP model [4], where amino acids are considered as single beads either hydrophobic (H) or hydrophilic (P), and protein conformations are embedded in a lattice as self-avoiding, connected paths. Given the number n_H of its hydrophobic (H-H) contacts between non-consecutive amino acids, the energy of a protein is typically defined to be $-n_H$.

1.2 Related Work

PSP in the HP model has been demonstrated to be an NP-hard problem [2]. Therefore, global optimization heuristics, and in particular evolutionary algorithms, have been reasonably used in order to solve it [5]. Unger and Moult [12] developed a genetic algorithm where only feasible (self-avoiding) conformations are allowed, and a Monte Carlo acceptance criterion biases crossover and mutation toward decreasing energies. The limits of this approach were discussed by Patton and colleagues [10], who also proposed a standard genetic algorithm based on the penalization of illegal (colliding) shapes. They designed a new conformational representation, called Preference Order Encoding, which further promotes feasibility within the population of solutions. Krasnogor et al. [5] analyzed three main factors affecting the efficacy of an evolutionary algorithm for PSP: the encoding scheme, the way illegal shapes are considered by the search, and the energy (fitness) function used. In the off-lattice, HP model of Piccolboni and Mauri [11], protein conformations are represented through a distance matrix, which was suggested to have considerable advantages over internal coordinates.

Section 2 describes in detail a memetic algorithm for PSP, while its results on a set of ten 48-mer sequences are presented in Section 3. A critical discussion of this work and its further developments is provided in Section 4.

2 Outline of the Approach

Memetic Algorithms (MAs) [8] are a powerful combination of local search with standard evolutionary algorithms. Their name stems from the term “meme”, which Richard Dawkins used in his book [3] to identify a cultural gene, evolving at a much faster pace than a natural gene. Central to MAs is the use of an improvement operator, based on local-search methods, alongside the standard “blind” mutation and recombination operators.

This paper presents a MA for PSP. We used a self-adaptive strategy, as the local search can act toward either exploitation or diversification of fitness, according to its degree of convergence within the population. Our algorithm is strongly based on a previous work of Krasnogor and Smith [6], where the authors successfully compared self-adaptation against other local-search approaches.

We describe now the implementation of our algorithm, with a number of graphs illustrating the main experiments we realized for tuning its parameters. All these graphs, except that of Fig. 4, show the average results of 30 independent MA runs. We always used the same 48-mer sequence (see below, Sequence no. 1 of Table 1), whose native structure has an energy of −32.

2.1 Encoding of Protein Structures

In order to find a convenient representation for lattice protein structures, we compared two integer-gene encodings on the basis of their impact on global search performance. Both use internal coordinates, i.e., they specify the sequence of directions that one must follow on the lattice to locate the successive amino acids of a polymer chain.

The first method is the classical relative-move encoding [10]. A widely known problem for this type of encoding is that, with increased chain length, randomly initialized populations tend to be dominated by infeasible shapes.

To overcome this problem, we decided to use the Preference Order Encoding (POE), introduced by Patton et al. [10].

2.2 Fitness

When PSP under the HP model is approached with evolutionary algorithms, it is natural to choose a fitness function that promotes, through its maximization, the formation of a hydrophobic core. Therefore, we selected a simple evaluation method that rewards H-H contacts and penalizes infeasible shapes. According to this method, the fitness f of a given protein structure s is determined as follows:

$$f(s) = n_H(s) - \rho \cdot n_L(s),$$

where $n_H(s)$ is the number of hydrophobic contacts in the protein, while $n_L(s)$ is the number of lattice sites occupied by two or more amino acids; the impact of the term $\rho \cdot n_L(s)$ tends to vanish under the POE, therefore ρ can be set to 1 for simplicity.

2.3 Genetic Operators

After some experiments, we chose 1-point crossover to be the recombination operator for our MA, as it proved to work better than both 2-point and uniform crossover.

Offspring mutation is applied at a very low rate, so it plays a minor role in the search process. When acting upon a protein's encoding, it blindly changes only one of its genes.

More clever mutation operators are used in the local-search process, and will be described in the following section.

2.4 Local Search

Our algorithm differs from a classical evolutionary algorithm in that the whole parent pool undergoes at each generation a step of local search (LS), which allows the algorithm to be classified as "memetic". We chose a self-adaptive strategy highly similar to the one proposed by Krasnogor and Smith [6], whereby the local search can be applied either to optimize or diversify fitness values, depending on their spread over the population (see Fig. 1 and 2). The probability of applying LS is kept fixed for the whole run, while the temperature parameter T is updated at each generation to the inverse of σ, the standard deviation of fitness within the population[1]. The temperature determines, according to a Boltzmann distribution, the chances of accepting a bad LS move, that is, a move causing a fitness decrease. So, when the population has a large variety of fitness values (high σ), the local search works toward optimization and bad moves tend to be rejected. Conversely, if fitness values become concentrated in a short range (low σ), much more bad moves are accepted and the local search acts as a diversification process. After some testing, the k coefficient was set to 0.96, a value relatively severe with bad moves; it remains to be studied to which extent this can be generalized to other problem instances.

```
LocalSearch(parents) {
    $T = 1/\sigma$;
    while($i \leq$ NUM_PARENTS) {
        /* $p_{LS}$: probability of applying local search */
        if(Random(0, 1) $< p_{LS}$) {
            LSmoves(parents[i]);
        }
        $i = i + 1$;
    }
}
```

Fig. 1. Pseudo-code for the LocalSearch function.

We designed a set of four LS moves, which are a 3D-cubic extension of those developed by Miller et al. [7] for a square-lattice model, and are implemented as macromutations of POE strings. As shown in Fig. 3, our moves result in at most two monomers shifting to a new lattice site, so conformational changes are made as local as possible; nonetheless, when a LS move leads to a collision, the POE implicitly forces a structural rearrangement that generally involves more amino acids [10]. Each time the Mutate function is called, a random extraction determines which of the four LS moves will be applied.

Testing these operators was an opportunity for verifying a sort of continuity of the fitness landscape: as illustrated in Fig. 4, little conformational changes, caused by LS moves, induce relatively small fitness variations; on the other hand, single-gene mutations, which imply the rotation of entire protein substructures, have also a higher impact on fitness.

[1] This technique is similar to the *reheating* proposed in [8].

```
LSmoves($ind$) {
    /* allow at most $MV$ consecutive moves */
    while($move < MV$) {
        $old = ind$;
        Mutate($ind$);
        $\Delta$ = Fitness($old$) − Fitness($ind$);
        /* $\Delta \leq 0$: always accept move
           $\Delta > 0$: conditionally accept move */
        if($\Delta > 0$) {
            if(Random(0, 1) $> e^{-k\Delta/T}$) {
                /* reject move and exit */
                $ind = old$;
                return;
            }
        }
        $move = move + 1$;
    }
}
```

Fig. 2. Pseudo-code for the LSmoves function.

Although the maximum number of chained moves MV was fixed at 5, our experimental results with a simple version of the MA indicate that local search ends, on average, after just 2.4 moves. This value is even lower, when the algorithm is enhanced with speciation mechanisms that promote genotypic diversity (see below). To confirm the advantages of LS hybridization over the classical approach [6], we compared the simple MA against a standard evolutionary algorithm without local search but with thrice as many generations. The results clearly encourage to use self-adaptive local search, which allows to find better solutions, while keeping a lower global convergence. In a similar experiment, our iterated-LS method outperformed the 1-step approach of Krasnogor and Smith [6]. Hence, investing most of the computation time in local search is an issue worth considering, when designing an evolutionary algorithm for HP-model PSP.

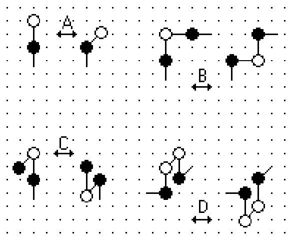

Fig. 3. The four LS moves available to the memetic algorithm. A: end move, B: corner move, C: end-corner move, D: crankshaft move. Shifting monomers are white, fixed monomers are black (adapted from Nunes et al. [9]).

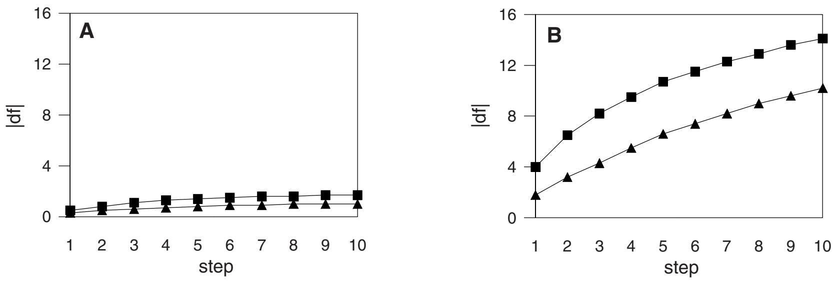

Fig. 4. Comparison between single-gene mutation (squares) and local-search moves (triangles), with respect to the fitness change they induce. A series of ten random perturbations was applied to a given protein structure and, at every step, the total fitness distance (|df|) from the initial conformation was measured. Each graph shows the average results of 1000 independent executions of this procedure. The starting proteins were drawn from a MA population, at the beginning of a run (A) and after 400 generations (B).

2.5 Speciation

While self-adaptive local search can, if needed, diversify the fitness in a population, it does little to avoid premature convergence at a genotypic level. Therefore, we believe the global evolutionary process may unduly get stuck in relatively narrow hyperplanes of the conformational space, which nonetheless guarantee a wide range of fitness values. In order to avoid such trapped conditions, we decided to promote speciation among protein structures by means of suitable selection, mating and replacement operators. They are all based on a simple scoring function, which estimates the degree of similarity between two conformations through a gene-by-gene comparison of their POE sequences. Fig. 5 presents the species-oriented selection, a two-step process which starts by structurally comparing a parent to a random set of other ones; that parent and its most similar individuals in the set, together forming a species, then undergo a classical tournament based on fitness, whose winner will be selected for reproduction. This procedure is repeated for all parents. Even simpler schemes allow the search to further emphasize speciation, through crossover of similar individuals and crowding-based replacement from the offspring population to the mating pool.

According to a great deal of experimental results, we can draw the following conclusions about the introduction of the speciation operators in the memetic algorithm: first, when species-selection is coupled with crowding-replacement, the search is undoubtedly more powerful (see Fig. 6), although both techniques, used alone, involve a clear deterioration of performance. Second, the size of the species set (SPC) in the selection function must be kept to 1, that is, each parent has to compete with just one rival, because a higher number would imply a premature convergence of the population. Third, by crossing similar individuals, the search seems to be unable to sustain the coevolution of separate species, even when

```
SpeciesSelection($par\_id$) {
    /* structural comparison. SMP = size of sample set */
    $sample\_set$ = RandomSet($parents$, $par\_id$, $SMP$);
    SortBySimilarity($sample\_set$, $par\_id$);

    /* fitness-based selection. SPC = size of species set. */
    $species\_set$ = SimilarityCut($sample\_set$, $SPC$);
    $winner$ = Tournament($species\_set$, $par\_id$);
    $mating\_pool[free\_slot]$ = $parents[winner]$;
}
```

Fig. 5. Pseudo-code for the SpeciesSelection function.

selection and replacement are biased toward speciation, for the average fitness rapidly converges to a poor local maximum, as if the population were dominated by a single genotype. An analogous behavior often concerns the replacement operator, when the crowding factor is increased beyond a value of 15–20: instead of obtaining, as expected, a higher spread of fitness, the net effect is to favor its global convergence. Therefore, the above results suggest to foster, but not to exasperate, speciation, although it would be useful to investigate in greater depth how speciation affects the distribution of genotypes in the population.

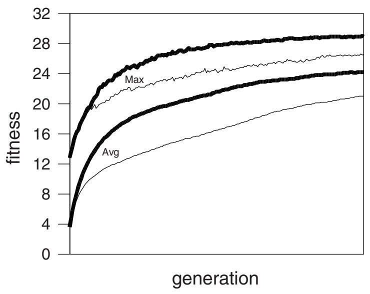

Fig. 6. Impact of speciation on search performance. A bold line represents a memetic algorithm where the operators of selection and replacement are species-oriented. A thin line refers to fitness-proportional selection and 100% replacement. The two topmost graphs plot the course of the maximum fitness, while the other two indicate the average fitness. Runs were 200-generation long. Note that, using standard operators, global convergence is lower, but increases at a much faster pace.

3 Experiments and Results

The experimental work described in Section 2, and many other tuning trials, led us to choose the following operators and parameters for what is, at the moment, the final configuration of our memetic algorithm:

- Preference Order Encoding
- same population size (1000) for parents, mating pool and offspring
- local search ($p_{LS} = 1.0$, MV=5, $k = 0.96$)
- unit penalty for each collided site ($\rho = 1$)
- intra-species selection ($SMP = 5$, $SPC = 1$)
- random mating
- 1-point crossover with probability 0.9
- single-gene mutation with probability 0.048 for each protein
- crowding replacement from offspring population to mating pool (step 1), with a crowding factor of 10
- 100% replacement from mating pool to parents (step 2)
- unit elitism of the best solution.

Table 1. The 48-mer test sequences with the lowest-energy conformations found for them by the MA (adapted from Yue et al. [13]). For all except proteins no. 7 and no. 9, actual native conformations are shown. Protein structures are represented by absolute-move encoding, which uses the directions of the lattice Cartesian axes (`F` = Forward, `B` = Backward, `L` = Left, `R` = Right, `U` = Up, `D` = Down).

1.	HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH
	RUURFLLDRDLLULBRDDBURUFUUFLDBUBDDBURFURRBDDFULD
2.	HHHHPHHPHHHHHPPHPPHHPPHPPPPPPHPPHPPPHPPHHPPHHHPH
	RDLDRBLUURDRRDLDFUUFUBRURBDLLUBDBDDFRULLBUFLDDR
3.	PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP
	RDRBLURFUBUULDFDLBRBRRUFFLFDRBRBLDRFLDDBUBUBLFD
4.	PHPHHPPHPHHHPPHHPHHPPPHHHHHPPHPHHPHPHPPPPHPPHPHP
	RURFUFDRBBBULDDDFRDBUUFLFDDFURBUFLLBLFDRBDBUBUB
5.	PPHPPPHPHHHHPPHHHHPHHPHHHPPHPHPHPPHPPPPPPHHPHHPH
	RBDFRBRBLBRRULLFFRBUFLBBBRFULLDLDRBDLFFURUURFLD
6.	HHHPPPHHPHPHHPHHPHHPHPPPPPPPHPHPPHPPPHPPHHHHHHPH
	RRULBDLLDFRBBRDLFFLBLBRDRRFURBRULBULFURDFDLFRDL
7.	PHPPPPHPHHHPHPHHHHPHHPHHPPPHPHPPPHHHPPHHPPHHPPPH
	RFURDBRBBLLDBRFFULDDFURFRBBDLDBLURBDRUFRULBBUFL
8.	PHHPHHHPHHHHPPHHHPPPPPPHPHHPPHHPHPPPHHPHPHPHHPPP
	RRBDLDDFURFFUBLUFFUBRDBRDDFDBLLULUBRRRDBLDFRBDD
9.	PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPHPHHHPPPPH
	RURRFDLBRBULUBDRDLFLDBBRFFDRURBLDLLLFRFRULLBUBR
10.	PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHHPPHHHHHHHPPHH
	RRDRBBLFULLBURDRRFRULBLUBLFUFDLDRFLURRDBURFFDBD

We then checked this algorithm with a set of ten 48-mer sequences previously used in a work of Yue et al. [13]. Such polymer chains are shown in Table 1, together with the absolute-move encoding [5] of the best conformations found for them by the MA. 30 runs of 6000 generations were performed for each sequence, with every single run taking about half an hour on a 1.8-GHz Linux PC. Table 2 presents the results of these test executions; since all best solutions are feasible, they are denoted by their HP-model energy, instead of their positive fitness.

As the reader can observe, our algorithm finds native conformations for eight of the ten sequences, and achieves nearly optimal solutions for the remaining two. This is an indication of a certain robustness, especially if we consider that all tuning tests were made uniquely with Sequence no. 1. Sequences no. 7 and no. 9 proved to be particularly difficult, because the algorithm never found any of their native states, even when running for 12000 generations.

The same table also shows no general correlation between the chances of finding native energies for a chain and its minimal native-state degeneracy (g_N), that is, a lower bound on the number of its native conformations [13]. This fact can suggest the existence of structural patterns very hard to form with the genetic operators we have designed so far: should these patterns be critical for the correct folding of a protein chain, it would be unlikely that the MA found its native structures.

Further results confirm that initialization actually contributes to the outcome of a search, although the best-fitness values fall in a short range. Hence, we recommend to distribute computational power among a number of independent runs, or, much harder, to design operators that guarantee a fair sampling of the solution space.

Table 2. Native energy (E_N), lowest energy of a MA solution (E_{MA}), number of successes out of 30 runs (n_{succ}) — i.e., number of runs where a native energy was found — and lower bound on native degeneracy (g_N) for the ten HP sequences shown in Table 1. Adapted from Yue et al. [13].

seq	E_N	E_{MA}	n_{succ}	g_N
1	-32	-32	7	1.5×10^6
2	-34	-34	1	14×10^3
3	-34	-34	1	5×10^3
4	-33	-33	4	62×10^3
5	-32	-32	7	54×10^3
6	-32	-32	3	52×10^3
7	-32	-31	0	59×10^3
8	-31	-31	2	306×10^3
9	-34	-33	0	10^3
10	-33	-33	2	188×10^3

4 Conclusions and Future Work

Although the above results are quite encouraging, we believe that providing the MA with more problem-specific knowledge would further enhance its search capabilities, with a minimal impact on robustness. For example, if we could figure out an even approximate picture of the energy landscape, based on the fraction or distribution of hydrophobic residues in a polymer sequence, a number of ad-hoc genetic operators, and an adaptive method to apply them, might be developed.

Making this MA, and evolutionary algorithms in general, perform true simulations of protein folding, would be another challenging problem. If that were possible, generations would represent the successive time-steps of a folding reaction, and their populations might indicate the most likely protein structures at those instants, from both a thermodynamic and kinetic point of view. To achieve this goal, genetic operators should be made aware of the particular folding phase they are applied in, so that their output conformations would not be out of the physical context. This way, evolutionary algorithms could represent a valid alternative to well established methods, such as Molecular Dynamics or Monte Carlo, for faithful simulations of protein folding.

References

1. C. B. Anfinsen, E. Haber, M. Sela, and F. H. White Jr. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. In *Proceedings of the National Academy of Sciences of the USA*, volume 47, pages 1309–1314, 1961.
2. B. Berger and T. Leight. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.
3. R. Dawkins. *The Selfish Gene*. Oxford University Press, New York, 1976.
4. K. A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501–1509, 1985.
5. N. Krasnogor, W. E. Hart, J. Smith, and D. A. Pelta. Protein structure prediction with evolutionary algorithms. In *GECCO '99: Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1596–1601. Morgan Kaufmann, 1999.
6. N. Krasnogor and J. Smith. A memetic algorithm with self-adaptive local search: TSP as a case study. In *GECCO 2000: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 987–994. Morgan Kaufmann, 2000.
7. R. Miller, C. A. Danko, M. J. Fasolka, A. C. Balazas, H. S. Chan, and K. A. Dill. Folding kinetics of proteins and copolymers. *Journal of Chemical Physics*, 96:768–780, 1992.
8. Pablo Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report C3P 826, California Institute of Technology, Pasadena, CA, 1989.
9. N. L. Nunes, K. Chen, and J. S. Hutchinson. Flexible lattice model to study protein folding. *Journal of Physical Chemistry*, 100(24):10443–10449, 1996.
10. A. L. Patton, W. F. Punch III, and E. D. Goodman. A standard GA approach to native protein conformation prediction. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 574–581. Morgan Kaufmann, 1995.
11. A. Piccolboni and G. Mauri. Application of evolutionary algorithms to protein folding prediction. In N. Kasabov, editor, *Proceedings of the ICONIP '97*, Berlin, 1998. Springer Verlag.
12. R. Unger and J. Moult. A genetic algorithm for 3D protein folding simulations. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 581–588. Morgan Kaufmann, 1993.
13. K. Yue, K. M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Shakhnovich, and K. A. Dill. A test of lattice protein folding algorithms. In *Proceedings of the National Academy of Sciences of the USA*, volume 92, pages 325–329, 1995.