STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 1, 2010

HILL-CLIMBING SEARCH IN EVOLUTIONARY MODELS FOR PROTEIN FOLDING SIMULATIONS

CAMELIA CHIRA

Abstract. Evolutionary algorithms and hill-climbing search models are investigated to address the protein structure prediction problem. This is a well-known NP-hard problem representing one of the most important and challenging problems in computational biology. The pull move operation is engaged as the main local search operator in several approaches to protein structure prediction. The considered approaches are: (i) a steepest-ascent hill-climbing search guided by pull move transformations, (ii) an evolutionary model with problem-specific crossover and pull move mutations, and (iii) an evolutionary algorithm based on hill-climbing search operators. Numerical experiments emphasize the advantages of the latter approach for several difficult protein benchmarks.

1. Introduction

Protein folding simulations aim to find minimum-energy protein structures starting from an initially unfolded chain of amino acids. The prediction of protein structures having minimum energies represents an NP-hard problem [1, 3]. The paper addresses this problem in the simplified hydrophobic-polar (HP) lattice model extensively engaged in computational experiments due to its simplicity [8], yet being able to generate significant results.

Several approaches to protein structure prediction based on evolutionary and/or hill-climbing search are investigated. The paper compares the perfromance of a pure hill-climbing search algorithm, a simple evolutionary algorithm and an evolutionary model based on hill-climbing search operators. The common feature of these approaches is the usage of pull move transformations [5] as the main local search operator. Pull move operations result in a single residue being moved diagonally causing the potential transition of

Received by the editors: December 5, 2009.

2010 *Mathematics Subject Classification.* 68T20.

1998 *CR Categories and Descriptors.* I.2.8 Computing Methodologies [**ARTIFICIAL INTELLIGENCE**]:Problem Solving, Control Methods, and Search – *Heuristic methods.*

Key words and phrases. evolutionary algorithms, hill-climbing search, protein structure prediction.

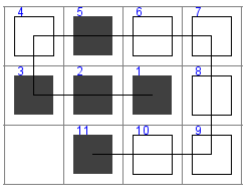

FIGURE 1. A protein configuration for sequence $SE = HHHPHPPPPPH$ in the square lattice having the energy value of -2. Black squares represent H residues and white squares are P residues.

connecting residues in the same direction in order to maintain a valid protein configuration [5]. Pull moves are engaged in different strategies in each model investigated.

Numerical experiments for bidimensional HP lattice protein sequences are carried out for all the models presented in the paper. Comparative results indicate a better performance of the evolutionary algorithm based on hill-climbing operators.

The paper is organised as follows: the protein structure prediction problem in the HP model is briefly described, pull move transformations are reviewed, the three models discussed in the paper are presented and numerical results and comparisons are given.

2. The Protein Structure Prediction Problem in the HP model

Simplified lattice protein models like the HP model [2] have become important tools for studying proteins being extremely useful in the initial approximation of the protein structure and in the investigation of protein folding dynamics.

In the HP model, a protein structure with n amino acids is viewed as a sequence $S = s_1...s_n$ where each residue $s_i, \forall i$ can be either H (hydrophobic or non-polar) or P (hydrophilic or polar). A valid protein configuration forms a self-avoiding path on a regular lattice with vertices labelled by amino acids. Figure 1 presents a configuration example for protein sequence $SE = HHHPHPPPPPH$ (black squares denote H residues and white squares represent P residues).

Two residues are considered topological neighbors if they are adjacent (either horizontally or vertically) in the lattice and not consecutive in the

sequence (for example in Figure 1 the pair of residues labelled 2 and 5 form a H-H topological contact).

In the HP model, the energy associated to a protein conformation takes into account every pair of H residues which are topological neighbors. Every H-H topological contact contributes -1 to the energy function. The aim is to find the protein configuration having minimum energy. This solution will correspond to the protein configuration with the maximal number of H-H topological contacts.

The energy of the protein conformation presented in Figure 1 is -2 (given by H-H contacts $2-5$ and $2-11$).

3. Pull Moves in the HP Square Lattice Model

Pull move transformations have been introduced in [5] as a local search strategy for the bidimensional HP model. Incorporated in a tabu search algorithm, pull moves have been able to detect new lowest energy configurations for large HP sequences having 85 and 100 amino-acids [5].

A pull move operation starts by moving a single residue diagonally to an available location. A valid configuration is maintained by pulling the chain along the same direction (not necessarily until the end of the chain is reached - a valid conformation can potentially be obtained sooner).

A pull move transformation can be applied at a given position i from the considered HP sequence.

Let (x_i, y_i) be the coordinates in the square lattice of residue i at time t. Let L denote a free location diagonally adjacent to (x_i, y_i) and adjacent (either horizontally or vertically) to (x_{i+1}, y_{i+1}). Location C denotes the fourth corner of the square formed by the three locations: L, (x_i, y_i) and (x_{i+1}, y_{i+1}). A pull move is possible if location C is free or equals (x_{i-1}, y_{i-1}). In the latter case, the pull move transformation consists of moving the residue from location (x_i, y_i) to location L. In the case that C is a free location, the first step is to move residue from position i to location L and the residue from position $(i-1)$ to location C. The pull move transformation continues by moving all residues from $(i-2)$ down to 1 two locations up the chain until a valid configuration is reached.

Figure 2 presents an example of a pull move transformation for HP sequence $SE = HHHPHPPPPPH$. The pull move is applied for residue H at position $i = 3$ for which a free location L horizontally adjacent to residue $i+1$ (between residues 4 and 10 in Figure 2.a) is identified. Location C (the location between residues 3 and 11 in Figure 2.a) is free in this example and therefore the pull move will cause moving the residue 3 to location L and

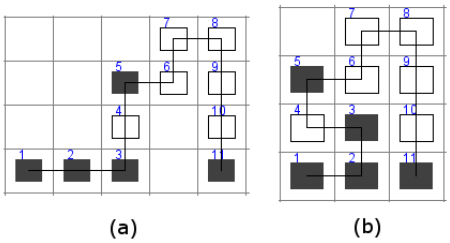

FIGURE 2. Pull move transformation for HP sequence $HHHPHPPPPPH$ at position 3

residue 2 to location C. The remaining residue 1 (only one in this example) is moved up the chain two positions (see Figure 2.b).

Lesh et al [5] prove that the class of pull moves is reversible and complete.

4. Evolutionary and Hill-Climbing Search

This paper investigates the performance of the following three models in solving the protein structure prediction problem: (i) a hill-climbing search algorithm, (ii) a simple evolutionary algorithm based on dynamic crossover and pull moves as mutation, and (iii) an evolutionary model based on hill-climbing search operators.

All three proposed models use the same problem representation (commonly engaged in genetic algorithms for this problem [7, 4]). A protein configuration (problem solution or chromosome) is encoded using an internal coordinates representation. For a protein HP sequence with n residues $S = s_1...s_n$, the chromosome length is $n-1$ and each position in the chromosome encodes the direction $L(Left)$, $U(Up)$, $R(Right)$ or $D(Down)$ towards the location of the current residue relative to the previous one. For the working example in Figure 1 the chromosome is $LLURRRDDLL$.

The fitness function used corresponds to the energy value of the protein configuration (as given in Section 2).

4.1. **Hill-Climbing Search Model.**

A simple hill-climbing search model based on pull moves is described (see *HC* scheme below). The algorithm applies pull move transformations for a protein configuration each iteration within a steepest ascent hill-climbing procedure.

Hill-climbing search starts by randomly generating one valid configuration for the given HP sequence and setting it as the *current_hilltop*. Pull moves are applied at each position $i, i = 1, ..., n$ (where n is the length of the HP

Hill-Climbing Search based on Pull Moves (HC)

```
Set current_hilltop to a randomly generated configuration rand_c
Set best_c to current_hilltop
Add best_c to hilltop_array
while (maximum number of hc iterations not reached) do
  for i=1 to n do
    Generate new configuration c_i by applying a
    pull move transformation at position i in current_hilltop
    if (c_i has better fitness than best_c) then
      Set best_c to c_i
    end if
  end for
  if (better configuration best_c found) then
    Set current_hilltop to best_c
  else
    Save best_c in hilltop_array
    Set rand_c to a new randomly generated configuration
    Set current_hilltop and best_c to rand_c
  end if
end while
Return best solution from hilltop_array
```

sequence) resulting in the generation of n new configurations. If any of them has a better fitness value than the *current_hilltop* it replaces the latter one. If no improvement is achieved and the maximum number of hill-climbing iterations has not been reached, the *current_hilltop* is saved in a list of hilltops and then reinitialized with a new randomly generated configuration.

4.2. **Evolutionary Algorithm with Pull Moves.** In the evolutionary approach (see *EA* scheme) to the protein structure prediction problem, a chromosome represents a possible protein configuration for a given HP sequence.

The population size is fixed and offspring are asynchronously inserted in the population replacing the worst parent within the same generation.

For the recombination of genetic material, a one-point dynamic crossover operator is specified. Given two parent chromosomes p_1 and p_2 and a randomly generated cut point χ, two offspring are created as follows. The genes before the crossover point χ are copied from one parent. The second part of the offspring is taken from the other parent in such a way that a valid configuration is maintained. This means that each position j, $j = \chi, .., n-1$ is copied from

Evolutionary Algorithm with Pull Moves (EA)

```
t = 0
Generate P(t) with pop_size individuals randomly
while (maximum number of generations not reached) do
  for each individual i in P(t) do
    Apply crossover with probability p_c
      Select mate j using binary tournament selection
      Generate random cut point χ
      Generate offspring o = crossover(i, j, χ)
      if o has better fitness than i or j then
        Replace worst(i, j) with o in P(t)
        Replace random individual from P(t) with mutation(o)
      end if
    Apply pull move mutation with probability p_m
      Generate random pull move position k
      Generate mutated chromosome m by pull move in i at position k
      if m has better fitness than i then
        Replace i with m in P(t)
      end if
  end for
  t = t + 1
end while
```

the second parent and checked for potential collisions with positions 0 to $j-1$ already copied in the chromosome current substring. If a conflict arises then a random direction leading to a valid position is selected and used in the offspring. The best of the two offspring generated replaces the worst parent if a better fitness was generated.

Pull move transformation is engaged as the mutation operator. For each individual selected for mutation, a random position is generated and a pull move transformation is applied at that position. The new mutated chromosome resulted replaces the parent if it has a better fitness value.

Furthermore, the offspring generated by crossover is transformed using pull move mutation and replaces an individual from the current population at random. This feature facilitates the diversification of genetic material and is also engaged in the evolutionary model with hill-climbing operators (presented in the following subsection).

4.3. **Evolutionary Model based on Hill-Climbing Operators.** In the third model investigated (see *EA-HCO* scheme), a population of configurations

Evolutionary Algorithm with Hill-Climbing Operators (EA-HCO)

$t = 0$
Generate $P(t)$ with *pop_size* individuals randomly
while (maximum number of generations not reached) **do**
  Randomly select p_1 and p_2 from $P(t)$
  while (maximum number of *hc* iterations not reached) **do**
    for $k = 1$ to n **do**
      Generate random cut point χ
      Generate offspring o_k = crossover(i, j, χ)
    end for
    Set o to $best(o_k)$, $k = 1..n$
    if o has better fitness than p_1 or p_2 **then**
      Replace $worst(p_1, p_2)$ - also in $P(t)$ - with o
      Replace random individual from $P(t)$ with $mutation(o)$
    else
      Set p_1 and p_2 to new randomly selected individuals from $P(t)$
    end if
  end while
  Hill-climbing pull move mutation for *hc* iterations
  $t = t + 1$
end while

is evolved by hill-climbing crossover and mutation. The evolutionary algorithm uses the same genetic operators described in section 4.2 (dynamic crossover and pull move mutation) with the difference that they are applied now in a steepest ascent hill-climbing manner.

Crossover is engaged for randomly selected pairs of individuals in a hill-climbing mode [6]. The best-fitted offspring replaces the worst parent within the same generation. If no better offspring is identified, both parents are replaced by new randomly selected chromosomes. The process continues until the maximum number of hill-climbing iterations is reached.

Mutation implements a steepest ascent hill-climbing procedure using the pull move operation. This process is able to generate a variable number of new individuals which replace parents within the same generation (if they have a better fitness value).

The hill-climbing pull move mutation step works in similar way with the procedure described in section 4.1 for hill-climbing search except that new

TABLE 1. Bidimensional HP instances used in experiments

Inst.	Size	Sequence	E^*
S1	20	1H 1P 1H 2P 2H 1P 1H 2P 1H 1P 2H 2P 1H 1P 1H	-9
S2	24	2H 2P 1H 2P 1H 2P 1H 2P 1H 2P 1H 2P 1H 2P 2H	-9
S3	25	2P 1H 2P 2H 4P 2H 4P 2H 4P 2H	-8
S4	36	3P 2H 2P 2H 5P 7H 2P 2H 4P 2H 2P 1H 2P	-14
S5	48	2P 1H 2P 2H 2P 2H 5P 10H 6P 2H 2P 2H 2P 1H 2P 5H	-23
S6	50	2H 1P 1H 1P 1H 1P 1H 1P 4H 1P 1H 3P 1H 3P 1H 4P 1H 3P 1H 3P 1H 1P 4H 1P 1H 1P 1H 1P 1H 1P 1H 1H	-21

individuals required for mutation are not generated anew but they are selected at random from the current population.

The number of individuals undergoing recombination and mutation each generation is dynamic as the hill-climbing operators modify the same structure until no further improvement can be generated and then continue with new individuals. An explicit selection for the next generation is not required as offspring are asynchronously inserted in the population as soon as they are created.

5. NUMERICAL EXPERIMENTS

The three models presented in the previous section are engaged in a set of numerical experiments for the bidimensional HP protein sequences presented in Table 1 (the known energy denoted by E^* is given for each instance).

The following parameter setting is engaged in the experiments:

- For the hill-climbing search model based on pull moves (refered to as *HC*), the number of *hc* iterations is 10000.
- For the evolutionary algorithm based on pull moves (refered to as *EA*), the population size is 100, the number of generations is 300, the crossover probability is 0.8 and the mutation probability is 0.2.
- For the evolutionary algorithm based on hill-climbing operators (refered to as *EA-HCO*), the population size is 100, the number of generations is 300, the offspring number in crossover hill-climbing is 50 and the number of hill-climbing iterations *hc* for both crossover and mutation is set to 100.

The initial population for the evolutionary algorithms contains randomly generated chromosomes representing valid configurations (each chromosome is

TABLE 2. Comparison of results achieved by the three investigated models for the HP problem

Inst.	Size	E^*	HC	EA	EA-HCO
S1	20	-9	**-9**	**-9**	**-9**
S2	24	-9	-8	**-9**	**-9**
S3	25	-8	-6	**-8**	**-8**
S4	36	-14	-10	-13	**-14**
S5	48	-23	-17	-20	**-23**
S6	50	-21	-15	-19	**-21**

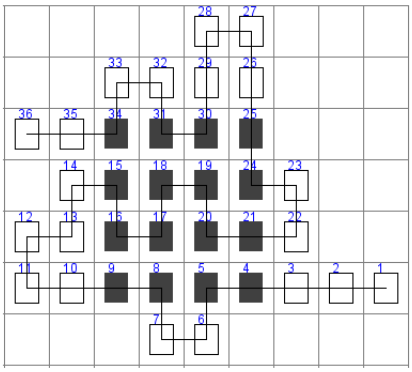

FIGURE 3. One of the protein configuration detected by *EA-HCO* for sequence S4 = 3P 2H 2P 2H 5P 7H 2P 2H 4P 2H 2P 1H 2P having the best-known energy value of -14

iteratively generated in a random manner until a conformation free of collisions in the HP square lattice model is found).

Table 2 presents comparative results for the HP sequences considered (the results of the best run out of 25 are reported). The known optimum energy E^* for each problem instance and the energy values detected by the three investigated models *HC*, *EA* and *EA-HCO* are given in separate columns.

Evolutionary search based on hill-climbing operators is able to detect optimal solutions for all HP instances considered. Figure 3 shows one of the optimal protein configurations detected by *EA-HCO* for instance *S*4.

TABLE 3. Percentage of succesful runs and the average generation number producing the best energy value for the *EA* and *EA-HCO* models

Inst.	EA			EA-HCO		
	E	Succ. Runs	Avg. Gen.	E	Succ. Runs	Avg. Gen.
S1	-9	92%	126.74	-9	100%	11.68
S2	-9	76%	153.26	-9	100%	16.20
S3	-8	64%	161.00	-8	100%	27.32
S4	-13	12%	235.00	-14	64%	141.68
S5	-20	4%	277.00	-23	8%	250.50
S6	-19	12%	221.33	-21	56%	182.07

Table 2 indicates that the evolutionary model based on hill-climbing search operators outperforms the other two approaches investigated. It can be observed that all three models are able to detect the best-known solution for the first sequence considered $S1$ having a length of 20. As the size of the protein sequence grows (and therefore the complexity of the search space increases), the power of hill-climbing search and evolutionary search alone gets lower.

Hill-climbing search (model *HC* in table 2) results are far from the optimum for the sequences $S2$ to $S6$ with lengths from 24 to 50. Evolutionary search (model *EA* in table 2) is able to identify optimum solutions for sequences $S1$, $S2$ and $S3$ but fails to guide the search towards the optimum for higher-size sequences. This problem is succesfullly overcome by the same operators applied in a hill-climbing manner in model *EA-HCO* - able to detect optimum energy values for all sequences considered.

The number of succesful runs (those in which the optimum energy has been detected) out of the 25 runs considered is studied in a further comparison between the *EA* and *EA-HCO* models. Moreover, the generation number producing the best energy value is recorded each run. Table 3 shows the results obtained in the following mode: for each HP sequence, the procentage of successful runs and the average generation number detecting an optimum (or best energy value) are given for the two evolutionary algorithms compared. It should be noted that table 3 considers succsesful runs those in which the best energy was obtained if the optimum was not found. This is the case of *EA* results for sequences $S4$, $S5$ and $S6$.

Table 3 clearly emphasizes the better performance of the *EA-HCO* model compared to *EA*. The percentage of successful runs is higher for each HP instance when the evolutionary algorithm based on hill-climbing search is used.

Furthermore, *EA-HCO* is able to detect the optimal solution in all 25 runs for several protein sequences. The *EA-HCO* model also outperforms *EA* with regard to the average generation in which the best energy configuration is identified. Hill-climbing search operators integrated in an evolutionary model are able to detect optimal solutions in the early stages of the search process. More generations are required as the protein sequence size increases. Even for such sequences, the *EA-HCO* is able to find the optimum solution earlier in the search compared to the stage where the *EA* model finds the best solution (not the optimum as *EA* fails to find optimum solutions for sequences $S4$, $S5$ and $S6$).

Numerical results and comparisons clearly emphasize the benefits of hill-climbing search operators integrated in evolutionary models compared to either hill-climbing or evolutionary search for protein structure prediction.

6. Conclusions and Future Work

Hill-climbing and evolutionary search models are studied for solving the protein structure prediction problem. The results presented emphasize the benefits of integrating hill-climbing search operators in an evolutionary algorithm.

Future work refers to the investigation of *EA-HCO* performance for other protein sequences and the extension of the proposed model to include other search operators.

Acknowledgments

This research is supported by CNCSIS grant PN II IDEI 508/2007 New Computational Paradigmes for Dynamic Complex Problems.

References

[1] Crescenzi, P., Goldman, D., Papadimitriou, C. H., Piccolboni, A., Yannakakis, M., *On the Complexity of Protein Folding*, Journal of Computational Biology, 50 (1998), 423–466.

[2] Dill, K.A., *Theory for the folding and stability of globular proteins*, Biochemistry, 24 (1985), 6, 1501–1509.

[3] Hart, W., Newman, A., *Protein Structure Prediction with Lattice Models*, Handbook of Computational Molecular Biology, Chapman & Hall CRC Computer and Information Science Series, 2006.

[4] Khimasia, M.M., Coveney, P.V., *Protein structure prediction as a hard optimization problem: the genetic algorithm approach*, Molecular Simulation, 19 1997), 205–226.

[5] Lesh, N., Mitzenmacher, M., Whitesides, S., *A complete and effective move set for simplified protein folding*, in RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology, ACM, 188–195 (2003).

[6] Lozano, M., Herrera, F., Krasnogor, N., Molina, D., *Real-coded memetic algorithms with crossover hill-climbing*, Evol. Comput., 12 (2004), 3, MIT Press, 273–302.
[7] Unger, R., Moult, J., *Genetic algorithms for protein folding simulations*, J. Molec. Biol., 231 (1993), 75–81.
[8] Zhao, X., *Advances on protein folding simulations based on the lattice HP models with natural computing*, Appl. Soft Comput., 8 (2008), 2, 1029–1040.

Department of Computer Science, Babes-Bolyai University, Kogalniceanu 1, 400084 Cluj-Napoca, Romania
E-mail address: cchira@cs.ubbcluj.ro