Tabu Search in Genetic Algorithm for Protein Folding Simulations in the 2D HP model

Alina-Gabriela Tunea¹

(1) University of Bucharest, Bucharest, Romania
alinaboca@yahoo.com

Abstract

One of the most important problems in computational molecular biology is to estimate the tertiary structure of a sequence of amino-acids for a protein.

In 1992 and 1993 the first articles were published containing the first demonstrations for the folded protein. Folded protein is a challenging problem of computing and is an NP-hard problem (NP-hard) even when the configuration is restricted to a grid (lattice). A successful method for solving this problem would have many implications in many areas including structural biology, genetics and medicine.

Programs are needed for efficient protein structure calculation. But even for simplified models using grids (lattices), the problem of predicting protein structure appeared to be NP-hard, and yet there is a polynomial algorithm

One of the most studied model abstractions of a protein structure is the hydrophobic polar model (HP) - Dill's model. Many algorithms have been made using 2D and 3D models in different varieties of grills.

The HP is based on the observation that hydrophobic forces are important forces for the problem of the folded protein.

In this model, a protein is represented as a linear sequence of amino acids consisting of two types: hydrophobic amino acids represented by H and hydrophilic or polar amino acids represented by P.

Calculating the functional structure of the amino acid structure is difficult as the space for possible structures is very large, making it difficult to search the “minimum energy” structure.

It was proposed a search method to find optimal conformation for two-dimensional HP models and also three-dimensional models.

Keywords: Protein folding, Tabu Search, Genetic algorithm

1. Introduction

For big complexity problems, finding the optimal solution or even an acceptable one could be a hard task. Classical techniques can’t be applied or they need long execution period of time.

Genetical algorithms are by definition searching stochastic algorithms based on natural principles of selection and recombination. Genetical algorithms are in fact strong searching techniques which are used with big success to resolve problems from many different disciplines "(Dumitrescu 2006)". Parallel genetical algorithms are very easy to be applied and they promise substantially good rates of performance "(Bălan and Pentiu 2009)". Genetic algorithms are efficient searching methods based on natural principles from genethics. They are applied successfully to get acceptable solutions to various business problems, engineering, science. Genetic algorithms represent, generally speaking, possibilities to find solutions in a reasonable period of time, but they are applied to problems that need increased periods of time to find adequate solutions "(Dumitrescu 2006)". As a consequence, there have been many efforts to make
faster genetical algorithms and one of the best choice is to use parallel implementations "(Bălan and Pentiuc 2009)".

Genetic algorithms are searching algorithms based on the natural selection mechanism. They are inspired by „survival of the fittest” principle, where the strongest individuals are selected after producing a new generation (offspring). In this context, individuals are candidate solutions to a given searching problem. This way, the reproduction of the strongest individuals is represented by solutions for the best current candidate solution. One of the advantages of a genetical algorithm versus the traditional methods is that it completes a global search using a population of individuals, more than a local search.

Genetic algorithms try to find optimal solution of the problem by manipulating a population of candidates’ solutions. Population is evaluated and the best solutions are selected for reproduction and mating in order to form the next generation. After some number of generations, the good traits dominate the population which results with more quality solutions.

The basic mechanism for genetical algorithms is Darwinistic evolution: the bad features are eliminated inside the population, because they appear for the persons that can’t survive selection process. The best surviving features are mixed in the recombination (mating) process to form individuals with better characteristics. The mutation also exists in the genetical algorithms domain but is considered a secondary operator. Its function is to assure that diversity is not lost for the population, this meaning that genetical algorithms can proceed to explore the solutions space “(Dumitrescu 2006)”..

2.1 Lattice Models
The well-known close form of the protein is made by the precise geometry of the inter-atomic contacts which stabilise the molecule: all possible interior links formed by hydrogen, and the part made by non-polar chains connected to form a tight interior package. The responsible forces for this precise geometry transform the folding chain into one rapid approximated form. The amount of energy of all molecules is time-consuming. One advantage is the possibility to start from a conformation almost correct "(Dill 1997, Liang and Wong 2001)".

Even if detailing the forces is very complicated, it has many potential applications. Thus, a good hierarchical approach could lead to the understanding and simulation of a biological process of assembling which is complicated "(Dill)".

The 2D square grids and 3D cube grids are the most studied grids and therefore they have exact methods of calculus, approximation algorithms and complex results. In the grid model, a protean sequence crease is defined by placing amino acids in the grid nodes according "(Dill 1997, Liang and Wong 2001)".

2.2 HP Model
In 1985, Ken Dill proposes the HP model (hydrofob-hydrofil), which is considered to have a fundamental role in the folded proteins modelling. According to this model, amino acids are classified as either H (hydrofob) or P (polar). Informally, a H & P sequence is integrated in a grid-type structure. A valid configuration corresponding to a sequence which auto-cancels into one grid. Using Lau and Dill’s terminology, we define as connected neighborhoods any 2 residuals k and k+1, which are adjacent to the given sequence, and also topological neighborhoods as adjacent residuals in a topological space (forming a contact), which aren’t also connected neighborhoods. The energy of the conformation can be calculated as the number of H-H contacts between topological neighborhoods "(Lesh, Mitzenmacher and Whiteside 2003)".
The 7th International Conference on Virtual Learning ICVL 2012

The figure shows a conformation with -2 energy (each H-H contact contributes with -1 to the total energy as long as other contacts don’t contribute). The black boxes represent hydrophobic residuals as long as uncoloured boxes represent polar residuals. Two hydrophobic contacts which contribute to the score are between 4th and 13th residuals and also between the 5th and the 12th sections.

Formally, for one sequence \( s \in \sum^n \) with \( \sum = \{H, P\} \) and \( n=|s| \), we define one conformation \( c_i \in C_s \) which has the energy \( E(c_i) \), where \( C_s \) is the lot of valid configurations for \( s \) and \( E(c_i) \) is the result of the following equation:

\[
E(c_i) = \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} N_{jk}, \quad \text{with}
\]

\[
N_{jk} = \begin{cases} 
-1 & \text{if j and k are both residuals} \\
0 & \text{else}
\end{cases}
\]

This model shows the fact that native folds of the protein tend to form very compact nucleuses, driven by dominant hydrophobic interactions. Every amino acid is classified as hydrophob (H) or hydrophil (P) and two hydrophobic amino acids are said to be in contact if they are adjacent in the fold but non-adjacent in the primary sequence. Taking into account that the objective is forming very compact hydrophobic nucleuses, optimization function is maximising the number of contacts between hydrophobic atoms (H-H contacts). For situations in which the problem is minimal energy, energy function is negative and it represents the number of hydrophobic contacts from the fold.

With this model we search for one configuration \( c^* \) which minimises the energy function \( E(c_i) \). Also, one configuration is considered a solution and named also basic configuration of a proteic sequence. Many instances of the HP problem present a degenerate solution, thus they have more than one minimal energy configuration. This way we define a basic configuration which doesn’t imply a unique solution, but one that is resolved by the following equation: \( E(c^*) = \min\{E(c_i) \mid C_i \in C_s\} \) (Unger and Moult 1993).

2.2 Protein Representation

The representation of a structure is in fact a valid configuration in a 2D grid. Each structure is represented as a sequence over the alphabet \( \Sigma = \{a, l, c\} \) where \( a \in \{H, P\} \), \( l \) is the line, \( c \) the column from the grid.

For example, for the following structure with 20 amino acid molecules with this form: HPHPPHHPHPHPHPH the sequence is represented like this: [H 2 4], [P 1 4], [H 1 3], [P 1 2], [P 2 2], [H 2 3], [H 3 3], [P 3 2], [H 4 2], [P 4 1], [P 5 1], [H 5 2], [P 5 3], [H 4 3], [H 4 4], [P 5 4], [P 5 5], [H 4 5], [P 5 5], [P 3 5], [H 3 4]

3. The genetic algorithm

One location is a node in a lattice corresponding to one pair of coordinates \((x,y)\). Locations are named adjacent if they are adjacent orizontally or vertically. Two locations are diagonally adjacent if these are at one orizontal step and another one step, vertically. One node from the chain is actually a molecule which has the H or P label. The nodes are numbered as consecutive from 1 to \( n \) all along the chain.
One mutation is a function which receives as input a valid configuration of the chain \( P(t) \) and produces a valid configuration \( P(t+1) \). One M mutations’ lot is reversible if for any move from M, applied to the \( P(t) \) configuration, results with the \( P(t+1) \) configuration. A moves’ lot M is complete if, with any configuration \( P \) and \( P^* \), there is a moves’ sequence M which moves (relocates) \( P \) to a congruent configuration (after translation or rotation) with \( P^* \).

Describing mutations is about valid configurations at different time periods. Location occupied by i node at t time is marked with \((x_i(t), y_i(t))\). A location is free at t time in the lattice if no other node is there.

Mutations applied to each chromosome are: rotations of 90 degrees, 180 degrees and 270 degrees, pull moves and tree bad flip.

Rotations. In order to apply the rotation operator, we select an amino acid with number k. The rest of amino acids, starting with \( k+1 \), will rotate around the amino acid on the k position(“Lesh, Mitzenmacher and Whiteside 2003”).

3.1 Tree bad flip moves
These moves are applied to the amino acids located in corners.

The sequence of 20 amino acids, presented up before mutation a) and after b). Amino acid no.12 was applied a tree bad flip mutation, the energy after mutation decreases from 0 to -2.

3.2 The pull moves
The pull moves were introduced by Neal Lesh. Let’s consider the i node at t time in the location \((x_i(t), y_i(t))\). Suppose that a free location L is adjacent to la \((x_i+1(t), y_i+1(t))\) and it has a diagonal adjacency at \((x_i(t), y_i(t))\). The \((x_i(t), y_i(t))\), \((x_i+1(t), y_i+1(t))\) nodes and free location L represent three corners of the square; the fourth one is location C. In order to make a Pull move, C location must be free or less equal with \((x_i-1(t), y_i-1(t))\). Local Pull move means movement of i node to L location. When C is free, the first node i is moved to location L and i-1 node is moved to C location. Then, until a valid configuration is ready, the following action is done: it starts with node \( j=i-2 \) down to node 1, it sets \((x_j(t+1), y_j(t+1)) = (x_j+2(t), y_j+2(t))\). The nodes are pulled two spaces up on the chain until a valid configuration is made. This guarantees the fact that a valid configuration is maintained: the nodes i and i-1 have been moved into a free location and the inferior indexed nodes are pulled repeatedly into a vacant location.

If pull moves go down to node 1, a valid configuration is ready. We can stop pull move of the inferior nodes when this happens. This improves moving point, as less nodes are changing position.

The sequence of 20 amino acids, presented up before mutation a) and after b). Amino acid no.6 was applied a pull moves mutation, the energy after mutation decreases from -2 to -6. For Example, “2.2 Main title” should be Times New Roman 11-point boldface, initially capitalised, flush left, with one blank line before, followed by your text on the next line. Use “Title Case” capitalisation.
3.3 The end moves

The end moves. For a n long chain, a final move can be done on the first residue or the nth. The residue is pivoted relatively to the neighborhood connected to a free adjacent position. This mechanism assures that the chain stays connected. If more than one valid position is free then one is chosen randomly.

The sequence of 20 amino acids, presented up before mutation a) and after b). Amino acid no.1 was applied a pull moves mutation, the energy after mutation decreases from -2 to -3.

3.4 The taboo search algorithm elements

The Tabu search examines the trajectory of a solutions’ sequence and it goes with the best neighbor of the current solution. In order to avoid the cycling, the solutions which were examined recently are forbidden, or tabu, for the given number of iterations.

The taboo list: A list of the latest obtained solutions. The movements memory can be a recent one or a frequency-based one. The recent movements made are stored into a mechanism which is referred as the Taboo-Move List. The number of moves in the list is determined by the dimension of the taboo list, marked as T. The list operates on the first-in-first-out principle. Other recent informations which are stored in the taboo-list are the configurations of the solutions.

The Taboo search solution is a solutions lot which have been created recently by the pull moves. The long-term memory, based on the frequency, allows for the search operations to be done in the best promising neighborhoods.

The candidates list: TS uses a candidate list, which offers a moves list to be evaluated. One move of the candidate list is chosen to continue the search. The list of candidates plays an important role in the TS performance.

Intensification and diversification: The intensification is in fact searching for good results from a recent found solution. The diversification encourages the search process to examine unvisited regions and to generate solutions which differ in many significant ways from previous solutions.

The termination criterion: The search is finished when the maximum number of iterations has been previously established, or after a predefined number of attempts have been made to set the same solution in the TS list as the new actual solutions "(Ji and Tang 2004)".

4. Procedure to apply mutation to Population

The procedure applies mutation to Population (generation_no), applies rotation mutations, Pull moves and tree_bad_flip to each position i from a cromosome and replaces initial cromosome with the best configuration obtained if fitness function is better than the one from the initial cromosome. Otherwise, we keep obtained configurations, for whom fitness functions obtained are better than the initial cromosome, into one table from where we can select next population.

4.1 The Tabu search algorithm based on Pull Moves and Rotations

noTransformation = number of Pull moves + number of rotations
best_fittest=fitness(c)
for i=1 to length (cromozom c) do
    for k=1 to noTransformation do
        Generate new configuration cik by applying a k pull move or rotation transformation at
        position i
        if cik has a best value fitness then
            best_fitnest=fitness (cik)
            save cik in tabu list array
        end if
    end for
end for

4.2 Procedure to apply crossover to Population

The cromosomes for whom fitness function has the best value are copied automatically into the
new generation and also are selected for crossover operation.

For crossover operation, cromosome selection is made after the selection probability.
Selection probability of the cromosomes ci is number pi given by:

\[ P_i = \frac{f(x_i)}{F}, i = 1, 2, ..., n \]

Because selection operator is applied n times, mean value of descendants from individual i is:

\[ n_i = n \cdot p_i \]

So: \( n_i = \frac{n \cdot f(x_i)}{\sum_{i=1}^{n} f(x_i)} \), where \( \sum_{i=1}^{n} f(x_i) \) is mean performance of the population.

After selecting the cromosomes, we will execute crossover operation.

SetOffspring= \( \emptyset \)

Procedure for applying crossover to Parents (generation_no) to produce Offspring (generation_no)
Select Parents (x,y)
for i=1 to length (cromosome) do
    Generate Offspring = crossover (x, y, i);
    if fitness value of Offspring is better than x and y then
        Add Offspring to SetOffspring
    end if
end for
if MaxOffspring is better of x and y then
    replaced worst (x; y);

After running the algorithm, we get the following results:

<table>
<thead>
<tr>
<th>Instance</th>
<th>Size</th>
<th>Sequence</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>20</td>
<td>1H 1P 1H 2P 2H 1P 1H 2P 1P 2H 1P 1H</td>
<td>-9</td>
</tr>
<tr>
<td>S2</td>
<td>24</td>
<td>2H 2P 1H 2P 1H 2P 1H 2P 1H 2P 1H 2P</td>
<td>-9</td>
</tr>
<tr>
<td>S3</td>
<td>25</td>
<td>2P 1H 2P 2H 4P 2H 4P 2H 4P 2H</td>
<td>-8</td>
</tr>
</tbody>
</table>
5. References


[12] Tianzi Jiang, Qinghua Cui, Guihua Shi, Songde Ma: Protein folding simulations of the hydrophobic–hydrophilic model by combining tabu search with genetic algorithms, In Journal Of Chemical Physics Volume 119, Number 8
