Record-to-Record Travel Algorithm for Biomolecules Structure Prediction

Ioan Sima(✉) and Daniela-Maria Cristea

Babes-Bolyai University, 400084 Cluj-Napoca, Romania
{sima.ioan,danielacristea}@cs.ubbcluj.ro,
http://www.cs.ubbcluj.ro/

Abstract. The biomolecules structure prediction problem (BSP) - especially the protein structure prediction (PSP) and the nucleic acids structure prediction - was introduced in the computational biology field approximately 45–50 years ago. The PSP on hydrophobic-polar lattice model (HP model) is a combinatorial optimisation problem, and consists in aims to minimize an arbitrary energy function associated with every native structure.

To solve the PSP problem, many metaheuristic methods were applied. Although the record-to-record travel algorithm (RRT) has proven useful in solving combinatorial optimisation problems, it has not been applied so far to solve the PSP problem.

In this paper, a mathematical modeling for PSP on the 2D HP rectangular lattice is developed and an adapted record-to-record travel algorithm (aRRT) is applied to address the combinatorial optimisation problem. For candidate solutions perturbation, a rotation and a diagonal move mutation operators were used.

A benchmark data set is used to test the RRT algorithm. The results obtained show that the algorithm is competitive when compared to the best published results. The main advantage of the RRT algorithm is that it is time-efficient, and requires small computational resources to obtain the same results as swarm intelligence algorithms.

Keywords: Record-to-record travel algorithm · Local search · Metaheuristic · Biomolecules · Protein structure prediction · HP lattice model

1 Introduction

Biomolecules are small or large molecules, being essential to biological processes present in living organisms. In chemical language, the large molecules are called macromolecules. All macromolecules are composed of tens to hundreds of thousands, even millions, of smaller subunits called monomers [8].

Macromolecules have several levels of structural organization. Biomacromolecular compounds such as proteins or nucleic acids (DNA and RNA) have four structure levels: primary, secondary, tertiary and quaternary structure,

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12949, pp. 449–464, 2021.
https://doi.org/10.1007/978-3-030-86653-2_33

respectively. In particular, all proteins reach the tertiary structure, but not all get to the quaternary structure. In the last two structure levels (tertiary or quaternary) the proteins and nucleic acids fold up to the functional state, called native structure (or folded conformation) [5].

Biomolecules structure prediction (BSP) consists in predicting the native structure based on the information stored in the primary structure. Myriads of tertiary conformations can be generated from only one primary structure. Only one conformation from this huge set of conformations is native conformation (i.e. it has the correct biological function). Even with today's most powerful computers, it is unfeasible in terms of time consumption to go through and check all the tertiary structures, even for very short macromolecules of dozens of monomers.

To address this problem by computational methods, a multitude of models have been proposed over time. One of the simplest models is hydrophobic-polar (HP), suitable for protein structure prediction problem (PSP), was presented in 1985 by Dill and developed in the next years [4,11]. It uses discretized spaces to represent monomers positions on the lattice. PSP on lattice model is a problem where a set of beans corresponding to the monomers of a protein must be arranged on a certain type of lattice. Even if this model is very simple, the exhaustive search in the combinatorial space is not feasible because PSP on HP model is proved to be NP-complete [2,3].

Consequently, a series of non-deterministic search techniques and metaheuristics have been applied to solving PSP so far. There were applied several classes of such techniques: evolutionary, metaheuristics, machine and deep learning, Monte-Carlo simulation and so on.

Among the first authors to address this problem were Unger and Moult, who applied genetic algorithm (GA) to the HP model [24]. GA-based researches continue to this day in a variety of variants. A combination of GA with Particle Swarm Optimization (PSO) was presented by Zhou [26], and Huang applied GA based on Optimal Secondary Structures (GAOSS) [10]. Recently, another combination of GA with Great Deluge Algorithm (GDA) proposed by Turabieh [23], and Evolutionary Monte Carlo (EMC) was applied on 2D HP by Liang and Wong [12]. Metaheuristic techniques used include Ant Colony Optimization Algorithm (ACO) [20], (PSO) [16], Ions Motion Optimization (IMO) algorithm combined with the Greedy Algorithm (IMOG) [25], etc.

Record-to-Record Travel algorithm (RRT) was introduced by Dueck in 1993 [6] as a variant of threshold accepting (TA) local search (LS) method [7]. It was applied with good results to many non-polynomial problems such as traveling salesman, knapsack, linear programming, quadratic programming, minimization of spin-glass Hamiltonian's, rough set theory [13,14], vehicle routing problem [19], etc.

This work applies an RRT algorithm to solve the PSP problem on the HP model. To our knowledge, although RRT is a fairly old algorithm, it has not yet been applied to this problem.

This paper is organized as follows. After the introduction, the optimisation problem formulation is given in Sect. 2. The third section explains the adapted Record-to-Record Travel algorithm. Section 4 contains the computational results and discussion, and the final section summarizes conclusions and further work.

2 Optimisation Problem

2.1 Biochemical Background

Biomolecules (or biological molecules) are molecules present in living organisms, frequently produced by them; they are essential to biological processes. Famous biomolecules include small molecules like amino acids (AA's), lipids, carbohydrates, nucleotides, antioxidants, polyols, vitamins, etc. and large macromolecules such as polynucleotides (DNA, RNA), proteins or polysaccharides [8].

All macromolecules are composed of tens to hundreds of thousands, even millions, of smaller subunits called monomers. For proteins, the monomers are the amino acids (20 types), for carbohydrates, the monomers are the monosaccharides, and nucleic acids (DNA, RNA) are build of the nucleotides. The proteins are organized in the form of complex structures that can be layered on four levels, called: primary, secondary, tertiary and quaternary structure. Protein primary structure is represented by the linear sequence (protein chain) of twenty types of amino acids. Secondary structures are represented by small local folding motifs, and tertiary structures of macromolecules are the three-dimensional shape of polymers [5]. This structure level is very important because proteins have biological function only in this form. The quaternary structure is a less common level, in which interactions between multiple 3D tertiary units occur.

The role of nucleic acids is to store genetic information, while proteins have multiple biological functions. Proteins are the molecules that provide life (the living). All other biomolecules participate in the proper functioning of the proteins. In conclusion, proteins are the most important constituents of biological life, hence the importance of knowing the native structure (folded) and the possibility of predicting it from the sequence of AA's (primary structure).

Biomolecules structure prediction consists of predicting the native structure based on the information stored in the primary structure, i.e. folded protein structure predicted from its linear sequence of amino acids.

For proteins, to one amino acids sequence (primary structure) there are many corresponding tertiary (or quaternary) structures. The problem is that the number of possible tertiary structures increases exponentially related to the length of the AA's sequence. Only one structure (called the native conformation) of this set of tertiary structures is physiological, aka is a "good" conformation. All other tertiary structures are pathological forms ("bad" conformations), causing various diseases [9]. The study of the protein folding phenomenon began in the '60s. Protein folding is the physical process by which a protein acquires its native conformation. As a result, researches on protein structure prediction are emerging in the next decade. The energy landscape theory of protein folding assumes that a protein's native state corresponds to its free energy minimum [1]. Figure 1

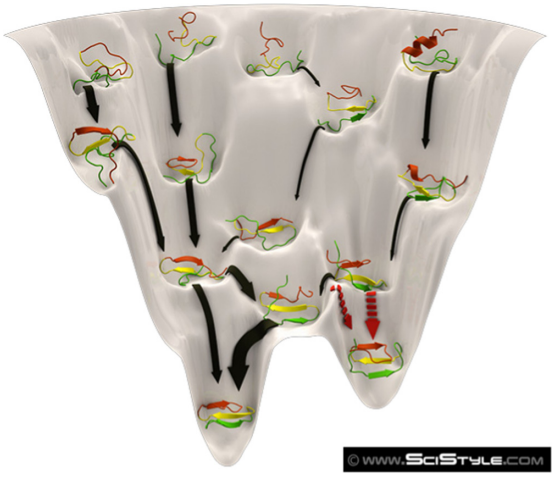

Fig. 1. An artistic representation of the funnel folding. The image was created by Thomas Splettstoesser [22]

shows the artistic image of the folding funnel, where many conformations fall down in local energy minimums. "Good" native conformation is found in global energy minimum.

Over time, starting with the '70s, to simplify and solve the PSP problem, various types of models have been proposed: all-atoms models, bean models, full energy models, statistical backbone potential models, lattice models, etc. [17].

2.2 The Hydrophobic-Polar Model

One of the simplest and well-known models for protein structure prediction is hydrophobic polar lattice model [11]. PSP on HP lattice is a discrete combinatorial optimisation problem. This model is a standard in testing the efficiency of the folding algorithms.

It introduces two main simplifications: i) the continuous Euclidean space is discretized, and every AA occupies one node of a specific regular lattice; ii) the 20 types of amino acids residues are reduced to two types of letters, based on their hydrophobicity:

1) "H": (Gly - G, Ala - A, Pro - P, Val - V, Leu - L, Ile - I, Met - M, Phe - F, Tyr - Y, Trp - W) and
2) "P": (Ser - S, Thr - T, Cys - C, Asn - N, Gln - Q, Lys - K, His - H, Arg - R, Asp - D, Glu - E) [18].

Note that, AA's can be denoted with 1-letter or 3-letter symbols. The "H" letter represents hydrophobic AA, and the "P" letter represents polar AA.

A *lattice*, Λ, in $\mathbb{R}^n$, is a set of n-dimensional vectors that form an additive group for any two points $\mathbf{u}, \mathbf{v} \in \Lambda$, i.e. it holds $(\mathbf{u} \pm \mathbf{v}) \in \Lambda$ [15].

A typical lattice Λ in $\mathbb{R}^n$ has the form:

$$\Lambda = \left\{ \sum_{i=1}^{n} b_i \mathbf{v}_i \;\middle|\; b_i \in \mathbb{Z} \right\} \tag{1}$$

where $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ is a basis for vector space $\mathbb{R}^n$.

Common lattices that have been studied in the context of lattice protein models are: 2D rectangular (in particular, 2D square), 2D triangular, 3D cubic and 3D face-centered-cubic (FCC) lattice.

At the primary structure level, a protein is represented by the ordered sequence Pr, of length n, where $Pr = (a_1, ..., a_n)$, and a_i is the ith AA in the sequence with $a_i \in \{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$.

In the HP model, the lattice coordinates define possible placements for AA's, and the primary structure of the protein will be represented by a string called *HP*, where $HP = (h_1, h_2, ..., h_n)$, and $h_i \in \{H, P\}$. The *HP* string is a constant because the order of the AA's in the primary structure of the protein sequence remains unchanged throughout the algorithm run. It can be put on the lattice, forming self-avoided walks (SAWs). A SAW is a sequence of moves on a lattice that does not visit the same point more than once. There are many ways to arrange an *HP* string to the lattice. Every path is encoded by a sequence of directions on a lattice, which is a string of length n or $n-1$. This string, which corresponds to a protein tertiary structure (rarely, to the quaternary structure), can be encoded in many ways: relative directions (*SRL string*; Straight, Right and Left), absolute directions (*RULD string*; Right, Up, Left and Down), binary representations, etc. All paths on the lattice must comply with the SAW rule.

Two adjacent letters of *HP* string are *sequence neighbours (or connected neighbours)*. Two non-adjacent letters of *HP* string, but which are neighbours after arranging on the lattice, are called *topological neighbours*.

In biological proteins, the folding process is strongly influenced by the hydrophobicity, i.e. they fold in the form of almost spherical globules with the hydrophobic amino acid residues in the protein kernel, and the polar ones, on the protein surface. To simulate this phenomenon, a distance-based arbitrary energy function is introduced. There are two types of letters in the *HP* string, thus, there are 2^2 types of contacts between topological neighbours. For every kind of contact, a value of free energy is introduced, as follows: e(H,H) = −1; e(H,P) = 0; e(H,P) = 0; e(P,P) = 0. Thus, H-H contacts are encouraged, and the others are neither encouraged nor penalized. A H-H contact (h_i, h_j) between the i-th and j-th AA from the original sequence can be formed iff $|i - j| > 2$.

The total free-energy, $E(c)$, of a walk (or conformation) c is computed as the number of H-H contacts, taken with the minus sign:

$$E(c) = - \sum_{i,j=1, |i-j|>2}^{n} e(a_i, a_j) \tag{2}$$

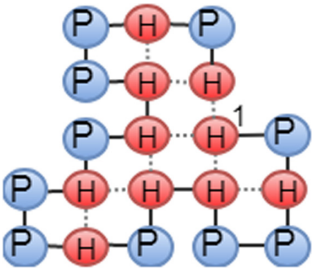

Fig. 2. Optimal conformation with the hydrophobic kernel (red) and polar surface (blue). Legend: *Hp* string (Input): HPHPPHHPHPPHPHHPPHPH. *RULD* string (Output): CRDDLULDLLURURULURRD. E(c) = −9 (number of dot lines). H - hydrophobic amino acid. P - polar amino acid. E(c) - energy of conformation (Color figure online)

In the optimisation approach, the conformation with the smallest energy has the greatest chance of forming a hydrophobic kernel (see Fig. 2); the search of native conformation from tertiary structures set is transformed into a search for the optimal walk on lattice (the SAW with the minimum total free-energy). In the HP model, the finding of native conformation in both 2D and 3D is NP-complete [2,3].

Table 1 summarizes the simplifications brought by the HP model and the correspondences to biological proteins.

Table 1. HP model summary

Biological proteins	Protein HP data structures
Primary structure	*HP string*
Tertiary structure (conformation)	*RULD string* - for absolute directions encoding
	SLR string - for relative directions encoding
Physical free-energy	Conventional free-energy
n - number of AA's	n - number of letter in HP string
Number of biological conformations $\geq 9^{n-1}$	Number of walks on 2D HP lattice: 4^{n-1} - absolute encoding 3^{n-1} - relative encoding
	Number of SAW: unknown

2.3 Mathematical Model for Protein Structure Prediction

In this work, the 2D HP square lattice, absolute encoding of directions and space coordinates of lattice positions have been used to denote the protein conformations (viz. SAWs).

The 2D square lattice is:

$$\Lambda = \{b_1\mathbf{v}_1 + b_2\mathbf{v}_2\} \tag{3}$$

where $b_1, b_2 \in \mathbb{Z}$, $\{\mathbf{v}_1, \mathbf{v}_2\}$ is a basis for vector space $\mathbb{R}^2$, and the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ meet the conditions: 1) $\mathbf{v}_1 \perp \mathbf{v}_2$, and 2) $\|\mathbf{v}_1\| = \|\mathbf{v}_2\|$. Thus, the primitive cell of a 2D HP lattice is a square whose sides, by convention, have length 1 (see Fig. 3).

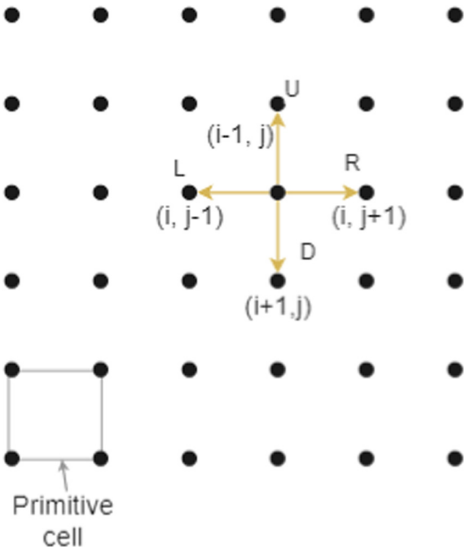

Fig. 3. The 2D square lattice

The main role of the lattice is to divide the continuous space in a discrete space. Thereby, amino acids spatial position is restricted to precise points in space.

For an *HP* sequence with n AA's, the $RULD$ is a string of length n, where $RULD = (d_1, d_2, ..., d_k, ..., d_n) \in \Lambda$, and $d_k \in \{C, R, U, L, D\}$. It represents a path on the 2D square lattice which must comply with the SAW restriction: in space coordinates, each $RULD$ letter is associated with a distinct 2D vector ($\forall k \neq l : \mathbf{v}_k \neq \mathbf{v}_l$). In 2D HP lattice, each node (point or vector) from a given position *(i, j)* has four neighbours, as you can see in Fig. 3. Hence, it follows that each AA can have a maximum of 4 AA's neighbours (extreme AA's in the *HP* sequence have one sequence neighbour and can have maximum three

topological neighbours, and the others have two sequence neighbours and can have maximum two topological neighbours).

Based on the minimum energy principle and the abstraction presented above, the following optimisation model (OM) for PSP problem on 2D square lattice can be established:

$$min \quad E(c) \tag{4}$$

subject to:

$$x^1_{n,n} = 1 \tag{5}$$

$$x^2_{n,n+1} = 1 \tag{6}$$

$$\sum_{k=1}^{n} \sum_{i=1}^{2n-1} \sum_{j=1}^{2n-1} x^k_{i,j} = n \tag{7}$$

$$0 \leq \sum_{k=1}^{n} x^k_{i,j} \leq 1, \quad \forall i, j \in \{1, 2, ..., 2n-1\} \tag{8}$$

$$x^{k+1}_{i,j} \leq x^k_{i,j-1} + x^k_{i,j+1} + x^k_{i-1,j} + x^k_{i+1,j} \tag{9}$$

where:

$E(c)$ is the free energy of *RULD* string c (protein conformation), $x^k_{i,j}$ is a three-dimensional variable where i and j are 2D vector coordinates of the lattice, and k is an kth AA in the *HP* sequence, such that:

$$x^k_{i,j} = \begin{cases} 1, & if \quad x^k = (i,j) \\ 0, & else \end{cases} \tag{10}$$

Here, "0" means that at position (i, j) there is no AA, and "1" means that at position (i, j) there is only one AA. Equation (5) fixes the first AA in the *HP* string at the position *(n, n)* on the lattice, and (6) fixes the second AA at the position *(n, n + 1)* on the lattice (right direction). This second constraint reduces the combinatorial space four times. Equations (7), (8) and (9) constrain that a walk in the grid occupies exactly n nodes, each lattice node cannot contain more than one AA and the two sequence neighbours AA's occupy the adjacent nodes in the lattice, respectively.

3 Record-to-Record Travel Algorithm

3.1 The Basic Algorithm

Record-to-Record Travel algorithm (RRT) as well as Great Deluge Algorithm (GDA), introduced by Dueck in 1993 [6], belong to the Threshold Accepting

algorithm (TA) [7]. All these methods (RRT, TA, GDA) are variants of Local Search (LS) metaheuristics. They are suitable for discrete optimisation; they are essentially *one-parameter* algorithms, meaning that it is necessary to tune only a single parameter, called *Deviation*.

The RRT and GDA algorithms are different from their predecessors, like Hill-Climbing or Simulated Annealing, in the acceptance way of a candidate solution from a neighbourhood. The RRT algorithm is very similar to the GDA algorithm, which is described by his author, Dueck, as follows: "Imagine, the GDA is to find the maximum point on a certain surface, for instance, the highest point in a fictitious empty country. Then, we let it rain without end in this country. The algorithm walks around in this country, but it never makes a step beyond the ever-increasing water level. And it rains and rains.... Our idea is that in the end the GDA gets wet feet when it has reached one of the very highest points in the country, so that is has found a point close to the optimum."

RRT differs from GDA by two points: i) RRT water level increase is given by the quality of the best solution found until that iteration; ii) a variable *Record* is updated only if the quality of the best solution is greater than its value.

To avoid the local minimum, RRT algorithm allows a poorer solution than the current solution ("submerged" solutions up to the value of the *Deviation* variable are accepted).

3.2 The Adapted Record-to-Record Travel Algorithm

In this paper, the RRT algorithm was adapted for PSP problem-solving. The description of adapted RRT includes three stages: representation and generation of the random initial conformation, evaluation of the conformations, and mutation.

Representing and Generating of the Random Initial Conformation. In this work, absolute directions encoding was used. The important data structures are: "*HP*" string, "*RULD*" string and "Chromosome" class, which encapsulates the two types of strings - a borrowed notion from the GA language. The *HP* string is the RRT input parameter, and the *RULD* string is the output one; both have the same length.

The generation of the first random solution (*RULD* string) starting from the *HP* string, is done through a stochastic backtracking variant in the same way as in our previous work [21]. For *HP* string (e.g. a chain of 6 AA's is: HHPHHP) it is built one *RULD* string (it can be CRRUUR). The first AA, "C", is placed in the center of the lattice, and the other AA's follow the next directions: right, right, up, up, and right. The first and the second letter ("CR") are always fixed. The next letter is chosen randomly from the {R, U, L, D} alphabet. Then the SAW condition is checked. If the point on the lattice is unoccupied, the next direction is generated and so on. Otherwise, the other letters of the alphabet {R, U, L, D} are chosen. If no direction passes the SAW rule, then it returns to the previous letter and the algorithm is resumed.

Evaluation of the Conformations. The energy of newly generated conformation is computed by:

$$E(c) = f(HPstring, RLUDstring) \tag{11}$$

Function f doesn't have an analytical formula. The algorithm counts the number of H-H contacts (topological neighbours) of the walk on the lattice. It depends on all the arguments shown in Eq. (11). In Fig. 2 is an example of optimum conformation well-arranged on the 2D HP lattice of an input *HP* strings with n = 20 AA's, the output *RULD* string, and the minimum energy.

Mutation. In the adapted RTT algorithm is implemented a k-point rotation and a diagonal move mutation operator. A conformation (or chromosome), $\mathbf{C} = \{d_1, d_2, ..., d_n\}$, where $d_i \in \{C, R, U, L, D\}$ is mutated to a new chromosome **C'**. The position g ($3 \leq g \leq n$), called mutation point, is random chosen in every iteration, and the letter at position g is replaced by one letter sampled uniformly from the {R, U, L, D} set. The number of mutation points in an iteration is equal with the value of k ($1 \leq k \leq n-2$). The letters at positions 1 and 2 (d_1 and d_2) remain unaltered by mutation because the first AA, C, is fixed in the lattice center, and the second AA is always R (positioned to the right relative to the first) because at the second position the other three directions (Up, Left or Down) give the same conformations, rotated by 90°, 180° and 270°, respectively. When $k = d$, the mutation produces a completely new random conformation, this way helping to explore the combinatorial space.

If the new letter, from position g, produces a feasible conformation, then the second part of the chain is rotated with 90°, 180° or 270°, respectively. The rotation occurs for k positions in each iteration.

The mutation operation (which is a small disturbance) can produce a conformation of quality close to the original conformation or one of very different quality, either extremely weak or optimal one. This is a way to see the chaotic behavior of the energy function on the HP model (Eq. 2). I.e., for sequence 1 from the data set, a mutation that substitutes U → R at the 12th position of a weak conformation (E = −4), CRDDLULDLLUULULDLUUR, produces an optimum conformation (E = −9), CRDDLULDLLURURULURRD (see Fig. 4).

In the second phase, the RRT algorithm searches the existence of one of the 16 *RULD* substrings that designate the square corner: UUR; UUL; DDR; DDL; RRU; RRD; LLU; LLD; URU; ULU; DRD; DLD; RUR; RDR; LUL; LDL. For the last substring found a diagonal move is executed, as shown in Fig. 5. If the new conformation is feasible, it is kept.

Finally, the mutation operation is:

$$\mathbf{C}(t+1) = Mutation(\mathbf{C}(t), g) \tag{12}$$

If the mutation operation fails (i.e. the resulting conformation is not feasible), a new letter is chosen from the directions not yet explored for the rotation mutation. If no variant is feasible, it proceeds to the next iteration and the old conformation is kept.

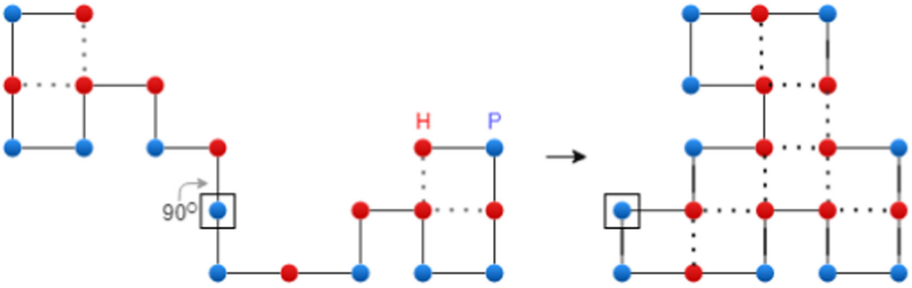

Fig. 4. The rotation mutation

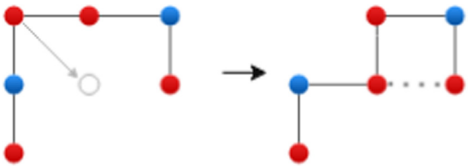

Fig. 5. The diagonal mutation

The Algorithm. After generating the first solution, its energy is computed, which is a negative value. The *Record* parameter gets this value. The *Deviation* parameter is set to a negative value close to zero. Afterward, for each iteration, a stochastic small perturbation is applied to the old configuration. If the energy of the new solution is better than the energy of the old solution, then the *Record* parameter is updated to the new energy. If the new energy is better than old energy or fall between old *Deviation* and *Record* parameters, then the new solution is accepted, even if it is worse solution. This step ensures the exploration of the solution space.

Because PSP on HP lattice is a minimization problem, the adapted RRT algorithm is mirrored, unlike the basic algorithm. Pseudo-code of the adapted RRT algorithm can be seen in Algorithm 1.

RRT algorithm works as follows: let **C** be the current solution with a *Record* equal to its free-energy and **C'** as an alternative solution derived from the first one through a small variation, like a mutation. The *Deviation* is set to a negative value close to zero. If the new free energy of **C'** is smaller than (*Record* - *Deviation*), then **C'** is the new solution in the iteration.

The main advantages of this algorithm are: it is time-efficient, requires small computational resources to obtain the same results as the other algorithms, and it has only two tuning parameters: *Deviation* and *MaxIter* (the number of iteration). Besides that, the RRT algorithm is easy to implement.

Algorithm 1: RRT algorithm

```
   Input  : MaxIter, HP sequence
   Output: C* (optimum RULD string)
1  t ⟵ 0;
2  Generate a random initial conformation, C(t) (a RULD string);
3  Update C* ⟵ C(t);
4  Compute Record ⟵ free-energy of the C(t);
5  Set an allowed Deviation ≤ 0;
6  while (t < MaxIter) do
       /* Mutation by eq (12)                                          */
7      if Exist(C ← Mutation(C(t))) then
8          Compute E ⟵ free-energy of new C;
9          if (E < (Record − Deviation)) then
10             Update C(t + 1) ⟵ C;
11         if (E < Record) then
12             Record ⟵ E;
13             Update C* ⟵ C;
14         t ⟵ t + 1;
15 return C*;
```

4 Computational Results and Discussion

So far, to our knowledge, the RRT algorithm was not applied to the PSP problem. We implemented the RRT algorithm in Python 3.7. and we ran the experiments on the next hardware configuration: Intel Core i5, 1.8 GHz CPU, 8 GB RAM under macOS Mojave (version 10.14) operating system.

The experiments were repeated 50 times independently for each HP sequence from the benchmark data set, and, for every experiment, were run 100,000 iterations. The *Deviation* parameter value is $Round(0.4 * MinEnergy)$ for every conformation. The MinEnergy is taken from literature.

The algorithm implementation will be publicly available at the https://github.com/simaioan/RRT_PSP

4.1 Benchmark Proteins Used in RRT Evaluation

The current RRT algorithm was applied for nine benchmark sequences taken from Unger [24] and Huang [10]. These benchmarks, shown in Table 2, are used to test RRT algorithm. The table contains information about sequence number, protein length (denotes the number of amino acids residues), the sequence of amino acids (H and P) and the best known free-energy (minimum or optimum).

4.2 Comparison to Other Methods

The RRT results are compared with the other four best optimal solutions obtained by other authors. Table 3 contains GA results taken from Unger and

Table 2. Benchmark data set

No seq	No of AA	Sequence	Optimal energy
1	20	HPHP PHHP HPPH PHHP PHPH	−9
2	24	HHPP HPPH PPHP PHPP HPPH PPHH	−9
3	25	PPHP PHHP PPPH HPPP PHHP PPPHH	−8
4	36	PPPH HPPH HPPP PPHH HHHH HPPH HPPP PHHP PHPP	−14
5	48	PPHP PHHP PHHP PPPP HHHH HHHH HHPP PPPP HHPP HHPP HPPH HHHH	−23
6	50	HHPH PHPH PHHH HPHP PPHP PPHP PPPH PPPH PPPH PHHH HPHP HPHPHH	−21
7	60	PPHH HPHH HHHH HHPP PHHH HHHH HHHP HPPP HHHH HHHH HHHH PPPP HHHH HHPH HPHP	−36
8	64	HHHH HHHH HHHH PHPH PPHH PPHH PPHP PHHP PHHP PHPP HHPP HHPP HPHP HHHH HHHH HHHH	−42
9	85	HHHH PPPP HHHH HHHH HHHH PPPP PPHH HHHH HHHH HHPP PHHH HHHH HHHH HPPP HHHH HHHH HHHH PPPH PPHH PPHH PPHPH	−52

Note: The 9th sequence (85 AA's) is taken from Huang [10].
The other sequences and optimal energy are taken from Unger [24].

Moult [24]; Genetic Algorithm based on Optimal Secondary Structures (GAOSS) taken from Huang C. et al. [10]; Evolutionary Monte Carlo (EMC) taken from Liang and Wong [12]; and Genetic Algorithm hybridized with Great Deluge Algorithm (GAGDA) which was taken from [23].

For relative short sequences ($n \leq 50$), RRT produces the best results (finds the conformation with the minimum free-energy known). For $n > 50$, RRT did not reach the optimal solution in the performed experiments. These results show that RRT requires improvements to achieve better results.

Table 3. Comparison of aRRT with other algorithms

No seq	Length	GA	GAOSS	EMC	GAGDA	aRRT
1	20	−9	−9	−9	−9	**−9**
2	24	−9	−9	−9	−9	**−9**
3	25	−8	−8	−8	−8	**−8**
4	36	−14	−14	−14	−14	**−14**
5	48	−22	−23	−23	−23	**−23**
6	50	−21	−21	−21	−21	**−21**
7	60	−34	−36	−35	−33	−31
8	64	−37	−42	−39	−42	−35
9	84	–	−52	−52	−52	−43

5 Conclusion and Further Work

The BSP is a combinatorial search problem. The PSP on HP lattice is a type of BSP that focuses on finding the most packaged protein conformations with the hydrophobic kernel.

A mathematical model was developed for PSP on the 2D HP lattice and a variant of local search, called Record-to-Record Travel algorithm was proposed and applied to predict the optimum protein conformations for nine benchmark sequences on a 2D HP square lattice, a well-known ab-initio model. The stochastic small perturbation of the old conformation was performed using rotation and diagonal move mutation operators.

Experimental results show that RRT algorithm finds the best solutions for relatively short sequences. For longer sequences, the results are promising. The interest is in the building of a more efficient mutation operator like crankshaft, rigid rotations, bead flip and so on. Further, experiments with other *Deviation* parameter values can be conducted to find better-energy.

Future researches will focus on the development of a population-based RRT algorithm (or GA hybridized with RRT) that can be applied to other lattice types: 2D triangular, 3D cubic, FCC, etc. The application of classic RRT and population-based RRT to other classes of biomolecules, like nucleic acids (DNA and RNA) is another proposed desideratum.

Acknowledgment. The authors would like to thank Prof. Univ. Dr. Bazil Parv for his professional guidance and valuable support.

References

1. Anfinsen, C.B.: Principles that govern the folding of protein chains. Science **181**(4096), 223–230 (1973)

2. Berger, B., Leighton, T.: Protein folding in the hydrophobic-hydrophilic (HP) model is np-complete. J. Comput. Biol. **5**(1), 27–40 (1998). https://doi.org/10.1089/cmb.1998.5.27, https://doi.org/10.1089/cmb.1998.5.27, pMID: 9541869
3. Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., Yannakakis, M.: On the complexity of protein folding. J. Comput. Biol. **5**(3), 423–465 (1998). https://doi.org/10.1089/cmb.1998.5.423, pMID: 9773342
4. Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry **24**, 1501 (1985)
5. Dinu, V., Trutia, E., Popa-Cristea, E., Popescu, A.: Biochimie medicală - mic tratat. Ed. Medicală, Bucureşti (2006)
6. Dueck, G.: New optimization heuristics: the great deluge algorithm and the record-to-record travel. J. Comput. Phys. **104**(1), 86–92 (1993). https://doi.org/10.1006/jcph.1993.1010, http://www.sciencedirect.com/science/article/pii/S0021999183710107
7. Dueck, G., Scheuer, T.: Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. J. Comput. Phys. **90**(1), 161–175 (1990). https://doi.org/10.1016/0021-9991(90)90201-B, http://www.sciencedirect.com/science/article/pii/002199919090201B
8. Garett, R.H., Grisham, C.M.: Biochemistry. Emily Barrosse, John J. Vondeling, 2nd edn. (1999)
9. Harrison, P.M., Chan, H.S., Prusiner, S.B., Cohen, F.E.: Thermodynamics of model prions and its implications for the problem of prion protein folding 11edited by a. r. fersht. J. Mol. Biol. **286**(2), 593–606 (1999). https://doi.org/10.1006/jmbi.1998.2497, http://www.sciencedirect.com/science/article/pii/S0022283698924974
10. Huang, C., Yang, X., He, Z.: Protein folding simulations of 2D HP model by the genetic algorithm based on optimal secondary structures. Comput. Biol. Chem. **34**(3), 137–142 (2010). https://doi.org/10.1016/j.compbiolchem.2010.04.002, http://www.sciencedirect.com/science/article/pii/S147692711000040X
11. Lau, K.F., Dill, K.A.: A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules **22**(10), 3986–3997 (1989). https://doi.org/10.1021/ma00200a030
12. Liang, F., Hung Wong, W.: Evolutionary Monte Carlo for protein folding simulations. J. Chem. Phys. **115**, 3374–3380 (2001). https://doi.org/10.1063/1.1387478
13. Mafarja, M., Abdullah, S.: A fuzzy record-to-record travel algorithm for solving rough set attribute reduction. Int. J. Syst. Sci. **46**, 1–10 (2015). https://doi.org/10.1080/00207721.2013.791000
14. Mafarja, M.M., Abdullah, S.: Record-to-record travel algorithm for attribute reduction in rough set theory. J. Theor. Appl. Inf. Technol. **49**, 507–513 (2013)
15. Mann, M., Backofen, R.: Exact methods for lattice protein models. BioAlgorithms Med-Syst. **10**(4), 213–225 (2014). https://doi.org/10.1515/bams-2014-0014, https://www.degruyter.com/view/journals/bams/10/4/article-p213.xml
16. Mansour, N., Kanj, F., Khachfe, H.: Particle swarm optimization approach for protein structure prediction in the 3D HP model. Interdiscip. Sci.: Comput. Life Sci. **4**(3), 190–200 (2012). https://doi.org/10.1007/s12539-012-0131-z
17. Neumaier, A.: Molecular modeling of proteins and mathematical prediction of protein structure. SIAM Rev. **39**(3), 407–460 (1997). https://doi.org/10.1137/S0036144594278060
18. Rashid, M.A., Khatib, F., Hoque, M.T., Sattar, A.: An enhanced genetic algorithm for ab initio protein structure prediction. IEEE Trans. Evol. Comput. **20**(4), 627–644 (2016)

19. Schittekat, P.: Deconstructing record-to-record travel for the capacitated vehicle routing problem. Oper. Res. Manag. Sci. Lett. **1**, 17–27 (2018)
20. Shmygelska, A., Hoos, H.H.: An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. BMC Bioinform. **6**(1), 30 (2005). https://doi.org/10.1186/1471-2105-6-30
21. Sima, I., Parv, B.: Protein folding simulation using combinatorial whale optimization algorithm. In: 2019 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 159–166 (2019)
22. Splettstoesser, T.: "© [Thomas Splettstoesser]/ Adobe Stock" Behance. Protein Folding Funnel. https://www.behance.net/gallery/10952399/Protein-Folding-Funnel. Accessed 02 Mar 2021
23. Turabieh, H.: A hybrid genetic algorithm for 2D protein folding simulations. Int. J. Comput. Appl. **139**, 38–43 (2016). https://doi.org/10.5120/ijca2016909127
24. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. J. Mol. Biol. **231**(1), 75–81 (1993). https://doi.org/10.1006/jmbi.1993.1258, http://www.sciencedirect.com/science/article/pii/S0022283683712581
25. Yang, C.H., Wu, K.C., Lin, Y.S., Chuang, L.Y., Chang, H.W.: Protein folding prediction in the HP model using ions motion optimization with a greedy algorithm. BioData Min. **11**(1), 17 (2018). https://doi.org/10.1186/s13040-018-0176-6
26. Zhou, C., Hou, C., Zhang, Q., Wei, X.: Enhanced hybrid search algorithm for protein structure prediction using the 3D-HP lattice model. J. Mol. Model. **19**(9), 3883–3891 (2013)