J Math Chem (2015) 53:794–806
DOI 10.1007/s10910-014-0459-1

ORIGINAL PAPER

An efficient approach for solving the HP protein folding problem based on UEGO

J. M. García-Martínez · E. M. Garzón · J. M. Cecilia · H. Pérez-Sánchez · P. M. Ortigosa

Received: 30 July 2014 / Accepted: 15 December 2014 / Published online: 24 December 2014

Abstract This work applies the methodology of the *Universal Evolutionary Global Optimization*, UEGO, to solve the protein structure optimization problem based on the HP model. The UEGO algorithm was initially designed to solve problems whose solutions were codified as real vectors. However, in this work the HP protein folding solutions have been defined as means of conformations encoded by relative coordinates. Consequently several main concepts in UEGO have been re-defined, i.e. the representation of a solution, the distance concept, the computation of a middle point, etc. In addition, a new efficient local optimizer has been designed based on the characteristics of the protein model. This work develops the adaptation and implementation of UEGO to the HP model and analyzes the UEGO solutions of HP protein folding for different 3D problems. Finally, obtained HP solutions are converted into all-atom models so that comparison with real proteins can be carried out, and a good agreement is obtained for small size proteins.

J. M. García-Martínez · E. M. Garzón · P. M. Ortigosa (✉)
Department of Informatics, ceiA3, University of Almería, Almería, Spain
e-mail: ortigosa@ual.es

J. M. García-Martínez
e-mail: chema.garc@gmail.com

E. M. Garzón
e-mail: gmartin@ual.es

J. M. Cecilia · H. Pérez-Sánchez
Bioinformatics and High Performance Computing Research Group (BIO-HPC), Computer Science Department, Universidad Católica San Antonio de Murcia (UCAM), Murcia, Spain
e-mail: jmcecilia@ucam.edu

H. Pérez-Sánchez
e-mail: hperez@ucam.edu

Keywords Protein folding · Evolutionary algorithm · HP model · Multiscale modeling · Algorithm acceleration

1 Introduction

Protein structure prediction and optimization is a well-known problem in structural bioinformatics. One of the most widely studied models of protein folding is the hydrophobic–hydrophilic (HP) model introduced by Dill et al. [9]. In the HP model, chains of amino acids are configured as self-avoiding walks on the 3D cubic lattice (i.e., adjacent amino acids of each chain lie on adjacent lattice sites, and no site is occupied by more than one amino acid). Based on the assumption that the hydrophobic interactions make an important contribution to the free energy of the folding process, a protein is modeled as a specific sequence of hydrophobic (H for nonpolar) or hydrophilic (P for polar) monomers. An optimal conformation maximizes the number of adjacencies between H's. So, this model is translated into an NP complete optimization problem, as shown in [4].

Due to the relevance of protein structure optimization and the effectiveness of the HP model, intensive research work in this line has been recently developed. Therefore, several approaches based on the application of different optimization methods are described in literature including Monte Carlo methods [30], evolutionary algorithms [28], ant colony optimization algorithms [11,22] and particle swarm optimization [16], just to name a few.

However, nowadays it is still an interesting challenge to design effective and efficient solutions of the HP protein folding problem and not only from the theoretical physics point of view but also because coarse grained solutions found by the HP model are in particular conditions close to the global minima of the protein folded structure and they can be coupled with an all-atom methodology, speeding up its convergence for finding representative solutions of the folding process.

In this work we explore the use of UEGO, a multimodal evolutionary algorithm [14,19], for solving the HP protein folding problem.

UEGO is able both to solve multimodal optimization problems where the objective function has multiple local optima and to discover the structure of these optima as well as the global optimum. UEGO has shown its effectiveness for solving different specific optimization problems [20,23,24,26]. Besides, it can be accelerated on several kinds of High Performance Computing Platforms [2,12,25] and its structure allows us to include specific local search procedures in order to improve the local search in particular scenarios. However, it is fair to underline that for solving a particular real problem by UEGO, such as HP Protein Folding, a twofold effort is necessary: (1) to re-define new main concepts based on the special codification and context of the particular problem and (2) to design an effective local optimizer.

This work is focused on the definition of the UEGO solution of HP Protein Folding. We show that this method achieves results comparable to other methods from literature at a much lower computational cost. Additionally, bearing in mind that UEGO is a multimodal global optimizer, it carries out a deep and wide exploration of the search space in order to provide all the local and global optima. Therefore UEGO presents

the advantage of exploring larger conformational spaces in the HP model and thus obtaining more optimal solutions than other reported optimization methods.

The paper is structured as follows; first a description of the HP model is introduced. Next, the main features of the UEGO evolutionary algorithm are explained and its implementation for solving the HP model is shown in detail. Next, a validation of UEGO in combination with an all-atom methodology using real protein structures is shown. Afterwards, results are obtained for different HP model benchmarks and results against other authors from literature are discussed in terms both of computational performance and ability to find lower scoring function minima. Finally, results comparing UEGO predictions against real protein structures are shown.

2 Objective function in HP protein folding problem

The HP protein folding model defined by Dill [9] has been widely used for predicting protein structures [3,6,18]. The HP model represents every protein sequence as a string $A = a_1a_2a_3 \ldots a_n$, where $a_i \in \{H, P\}$ and $1 \leq i \leq n$. A conformation of A is defined by a sequence of fold directions starting from the lattice site occupied by the first amino acid residue a_1. The protein conformations are restricted to self-avoiding paths on D-dimensional sequence lattice, where $D = 2$ or 3. Most protein structure prediction methodologies assume that the native state of the protein is defined by the lowest value of the Gibbs free energy what is estimated by a specific scoring function, which depends strongly on the coarse-grained or all-atom model used for representing the protein structure [10]. In the HP model, the energy of a conformation is defined by a scoring function that takes into account the number of topological contacts between hydrophobic amino acids that are not neighbours in the given sequence.

Thus, the protein structure prediction is translated into an optimization problem as follows:

Given an amino acid sequence $A = a_1a_2a_3 \ldots a_n$ find an energy minimizing conformation $\mathcal{C}^o$.

$$E(\mathcal{C}^o) = min\{E(\mathcal{C}) \forall \mathcal{C}\} \tag{1}$$

where $\mathcal{C}$ is a possible conformation of the string A and $E(\mathcal{C})$ is defined by

$$E(\mathcal{C}) = \sum_{i,j} e(a_i, a_j) \tag{2}$$

where

$$e(a_i, a_j) = \begin{cases} -1 & \text{if } a_i, a_j = HH \text{ and they form a topological contact;} \\ 0 & \text{in other cases.} \end{cases}$$

Therefore, the objective function for the HP protein folding problem is defined by Eq. (2) and its values are also referred to as scoring values. The goal of UEGO is the computation of the conformations which achieve minimum values of the energy or scoring function according to Eq. (2).

3 UEGO for solving HP protein folding problem

UEGO [14,19] is a multimodal algorithm which is able both to solve multimodal optimization problems where the objective function has multiple local optima and to discover the structure of these optima as well as the global optimum (see Algorithm 1 for a global description of UEGO). This algorithm is a general purpose metaheuristic, which can be tuned to deal with many problems. In particular, it may make use of the knowledge of a given problem by including a specific local optimizer in its structure. In fact, UEGO is an iterative process, which guides the local optimizer in a search for feasible solutions. UEGO works with a population of individuals. The use of a population ensures the exploration of the search space, while the use of local search techniques helps to quickly identify "good" areas in the search space. This provides a balance between the *exploitation* of the accumulated search experience and the *exploration* of the search space to identify regions with high quality solutions.

3.1 Basic concepts in UEGO

A key notion in UEGO is that of a species s. A species would be equivalent to a subpopulation in a usual multimodal evolutionary algorithm. A species can be thought of as a window (sphere) on the whole search space. This window is defined by its center C and a radius r_i that is associated to the level or iteration i in which the species has been created. The center is a solution, and the radius is a positive number that indicates the center attraction area which covers a region of the search space and hence, multiple solutions. It is interesting to remark that this definition assumes a *distance* defined over the search space.

The role of this window is to 'localize' the optimizer that is always called by a species and can 'see' only its window, so every new sample is taken from there. The radius of a species is not arbitrary; it is taken from a list of decreasing radii, the radius list that follows a cooling schedule, in such a way that given the smallest radius and the largest one (r_l and r_1) the remaining radii are expressed by a decreasing exponential function.

Algorithm 1 describes the UEGO basic structure [14]. Initially, a species or population of a single randomly generated individual is created. Once the initial population is obtained, an iterative process is carried out. At each iteration, a new offspring is generated using recombination operators and a selection procedure. To increase the fitness of individuals a local optimizer is then applied.

In UEGO the most important parameters are those defined at each level: the radii (r_i) and the function evaluation numbers for species creation (new_i) and optimization (n_i). These parameters are computed from some user-given parameters that are easier to understand.

The maximum length of the species list is given by the input parameter M and the number of existing species at level i is denoted by S_i.

The maximum number of function evaluations for the whole optimization process is given by the input parameter N. In addition, each level i has two restrictions on the

Algorithm 1 The **UEGO** algorithm

```
1: Init_species_list
2: Optimize_species(n_1)
3: for i = 2 to levels do
4:     Determine r_i, new_i, n_i
5:     Create_species(new_i/S_i)
6:     Fuse_species(r_i)
7:     Shorten_species_list(M)
8:     Local_Optimize_species(n_i/M)
9:     Fuse_species(r_i)
10:    Shorten_species_list(M)
11: end for
```

number of function evaluations (f.e.), namely new_i (maximum f.e. allowed when creating new species) and n_i (maximum f.e. allowed when optimizing individual species). These parameters are related through: $N = \sum_{i=1}^{levels}(n_i + new_i)$. For further details about the mathematical expressions of n_i and new_i see [14].

3.2 Adaptation of UEGO to HP protein folding problem

3.2.1 Codification of the problem

In the HP protein folding problem, the main interest is focused on the set of particular folding conformations of a protein sequence which define the search space. In [4,11] these conformations are defined by a graph which will primarily be the 3-dimensional cubic lattice Z^3. Then, a particular folding is defined as an injective mapping from $[1, \ldots, n]$ to the graph so that adjacent integers map to adjacent nodes on the graph. This definition of conformation is focused on the spatial location of a protein sequence. It is very suitable to evaluate the energy of every conformation which is related to the number of topological contacts between hydrophobic amino acids that are not neighbors in the sequence. In this context the distance between conformations' based on the spatial locations of amino acid can be well defined in relation to the Euclidean space. However, it is computationally very expensive due to the necessary alignment process.

An alternative codification of folding conformations is also defined in literature [17,29,32]. It is based on relative coordinates which define the location of every amino acid in relation to its contiguous elements in the protein. More specifically, let A be a protein sequence of length n, then conformations are defined by strings of length $n-1$ over the symbols $\{r(ight), l(eft), f(orward), d(own), u(p)\}$, and that denotes a valid conformation in the 3D square lattice [17]. Notice that the symbol $b(ack)$ is not included in the conformation strings because the reference system for the relative coordinates is defined by the folding address and location of each amino acid, so 'back' folding is not allowed in this codification of conformations.

In this work UEGO has been adapted to the HP protein folding problem by means of conformations encoded by relative coordinates. So, the search space consists of the set of strings of length $n-1$ over the symbols $\{r, l, f, d, u\}$.

3.2.2 Definition of a distance function

It is relevant to underline that the distance between two elements into search space is the key of the UEGO computation. In this context, it is possible to define a function of two conformations, $\mathbf{x} = \{x_i\}$ and $\mathbf{y} = \{y_i\}$ where $1 \leq i \leq n$ and $x_i, y_i \in \{r, l, f, d, u\}$ as the number of different elements in both sequences and it can be mathematically expressed as follows:

$$d(\mathbf{x}, \mathbf{y}) = \mid \{i \in [1, n] / x_i \neq y_i\} \mid \tag{3}$$

where $\mid \{...\} \mid$ represents the number of elements of the set defined in the brackets. This way, $d(\mathbf{x}, \mathbf{y})$ is defined as the number of different elements in both sequences, $\mathbf{x}$ and $\mathbf{y}$. It can serve as proof that function $d(\mathbf{x}, \mathbf{y})$ is a metric or distance integer in the search space because it satisfies the four conditions for a metric [1]:

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$;
2. $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$;
3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
4. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ (triangle inequality).

Proofs of 1–3 are trivial according to definition (3).

Proof of triangle inequality: If $d_i(\mathbf{x}, \mathbf{y})$ is defined as the contribution of elements x_i and y_i to $d(\mathbf{x}, \mathbf{y})$, the following options are possible $\forall i \in [1, n] \subset Z$:

1. if $x_i = y_i = z_i$ then $d_i(\mathbf{x}, \mathbf{y}) = d_i(\mathbf{x}, \mathbf{z}) + d_i(\mathbf{z}, \mathbf{y})$;
2. if $x_i = y_i \neq z_i$ then $d_i(\mathbf{x}, \mathbf{y}) < d_i(\mathbf{x}, \mathbf{z}) + d_i(\mathbf{z}, \mathbf{y})$;
3. if $x_i = z_i \neq y_i$ or $y_i = z_i \neq x_i$ then $d_i(\mathbf{x}, \mathbf{y}) = d_i(\mathbf{x}, \mathbf{z}) + d_i(\mathbf{z}, \mathbf{y})$;
4. if $x_i \neq y_i \neq z_i$ then $d_i(\mathbf{x}, \mathbf{y}) < d_i(\mathbf{x}, \mathbf{z}) + d_i(\mathbf{z}, \mathbf{y})$;

According to the accumulative contribution of the i-th elements of conformations to the distances, it can be concluded that triangle inequality is verified for the metric defined by (3). Therefore, this metric allows us to compare the folding of two conformations by a light computation on their relative coordinates.

3.2.3 Adaptation of UEGO procedures

UEGO is used for solving the folding problem according to this definition of distance. UEGO is based on the concept of species defined by a center C and a radius r. Therefore, according to the distance above defined, a species represents the set of protein conformations which share at least $n - r$ relative coordinates with its central conformation C.

The UEGO computation is an iterative procedure composed of the stages defined in Algorithm 1. Next, every UEGO stage is described according to the the concept of species and distance related to the set of possible conformations of a specific protein.

$Create\_species(new_i / S_i)$ This procedure generates a random sampling within every species, and for every pair of the previously generated random points a third point is computed in the middle of both points. The random conformations are generated by the random selection of $k_x, k_y \leq r$ elements of C to be randomly updated. The generation of the middle point is not direct because the search space of conformations is not defined as a vectorial space. However, from two conformations $\mathbf{x}$ and $\mathbf{y}$ with

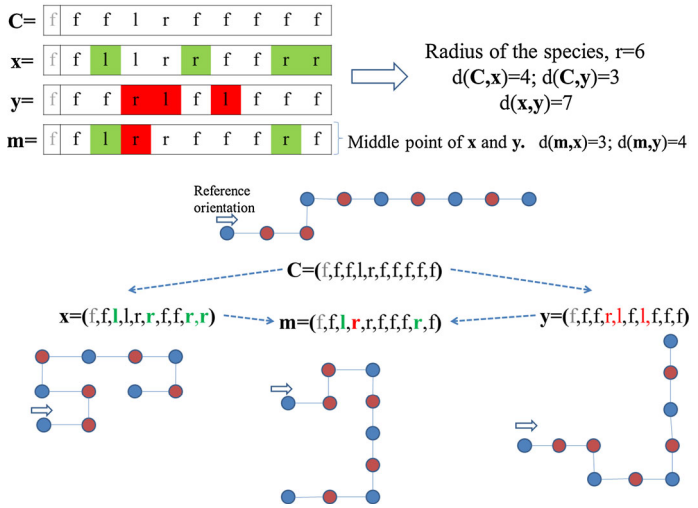

Fig. 1 Illustration of the sampling approach in the Create_species procedure for a 2D protein of $n = 10$: (*Top*) Generation of two random points, **x**, **y** and the middle point **m** in one species with center **C** and radius $r = 6$. (*Button*) Spatial representation of the conformations related to the *points on the top*

$d(\mathbf{x}, \mathbf{y}) \leq k_x + k_y$ it is possible to generate a new conformation, **m**, so that $d(\mathbf{x}, \mathbf{m}) \approx d(\mathbf{y}, \mathbf{m}) \approx (k_x + k_y)/2$. To compute the middle conformation, **m**, a new conformation is defined by the combination of both, **x** and **y**.

Figure 1 illustrates this general scheme to generate **m** from **x** and **y**. This way the new conformation **m** can be considered a middle point of the pair **x** and **y**. Nevertheless, the similarity among $d(\mathbf{x}, \mathbf{m})$, $d(\mathbf{y}, \mathbf{m})$ and $(k_x + k_y)/2$ is approximate, since the integer distance and the general conformation restrictions can change the general scheme to define several coordinates of **m**.

To complete the computation of *Create_species* the energy or scoring function (Eq. 2) is evaluated at the random pairs of conformations **x** and **y** and the middle points **m**. Once the points are evaluated:

- if **m** has a larger scoring value than the members of the pair **x** and **y**, then both members are considered to belong to different species and so, two new species are created. Their centers are defined by **x** and **y** and the associated radius is r_i, the one corresponding to level i.
- if **m** does not have a larger scoring value than the members of the pair **x** and **y** the center of the calling species is replaced by any computed point that presents a smaller scoring value.

$Fuse\_species(r_i)$ Whenever the centers of any pair of species are closer to each other than the given radius (r_i), the two species are fused. The center of the new species will be the one with the better energy value while the level will be the minimum of the levels of the original species (so the radius will be the larger one).

$Shorten\_species\_list(M)$ It only consists of deleting the latest species whenever the total number of species is greater than the maximum allowed.

$Local\_Optimize\_species(n_i/M)$

As can be seen in Algorithm 1, UEGO is a hybrid algorithm that introduces a local optimizer into the evolution process (*Local_Optimize_species*). In this way, at every generation, UEGO performs a local optimizer operation on each species, and these locally optimal solutions replace the caller species. UEGO is abstract in the sense that the 'species-management' and the cooling mechanism have been logically separated from the actual local optimization algorithm. Therefore, it is possible to implement any kind of optimizer to work inside a species.

Each solution in the HP protein folding problem has a great amount of restrictions within the allowed foldings, and therefore is rather difficult using a general local optimizer to find improved local solutions. As a consequence, the use of specific local optimizers in which some prior knowledge of the problem and some physical forces have been taken into account is more efficient.

Three different local optimizers have been considered in this work, i.e. *zip* based on the zipping and assembly (ZA) search strategy described in [21], *quake* and *shake* strategies referenced in [15] as "recipes" provided by different users that help to create accurate protein structure models. The term "recipes" is inspired by the design system folding algorithms used in the on-line "fold-it" platform where players can design methods for folding based on the programming language LUA proteins. According to a study published by [21] a number of algorithms have been highlighted for the prediction of protein folding. These algorithms have been combined in recipes by selecting specific levels of performance of the algorithm and the have obtained very good results.

These local optimizers or recipes address the search of an optimal folding following different strategies among them. We have carried out a preliminar study considering different local optimizers in order to find the one that obtains better results. In this preliminary study, for each local optimizer, a multistart optimization algorithm has been implemented where this specific local optimizer was applied to several initial solutions randomly generated. In all the cases, the results provided by these multi-start algorithms were not good enough, and therefore, after several experiments with different local optimizers we decided that using a single local optimizer was not the best option as none of them presented relatively good results. As a consequence we defined as local optimizer for UEGO a search strategy that consists of a mix of the three local optimizers metioned above, i.e. *zip*, *quake* and *shake*. The structure of the local optimizer implemented is described in Algorithm 2. As can be seen variants turnaround folding have also been added in some of the recipes because it favors the exploration of the bending sequence in reverse to the trend that was initially created this sense sequence. An additional element is introduced tracking that can extend at times the search space.

The input parameter of this procedure is the number of function evaluations devoted to local optimization n_i at current level i divided by the number of existing species (S) at that level.

To apply local optimization on each species, a multi-heuristic optimizer has been developed. It is able to apply six different transformations to the folding sequence where a mix of accurate and random heuristic criteria for changes in the sequence folding has been applied. These heuristics are combined into "recipes" that have the same probability of execution. At each iteration of Algorithm 2 a recipe or local

Algorithm 2 The **Local_Optimize_species**(n_i/S_i) algorithm

```
for j = 1 to S_i do
    for k = 1 to n_i/S_i do
        H = random(1,4)
        Case H
          1: S*_j = LO1(s_j)
          2: S*_j = LO2(s_j)
          3: S*_j = LO3(s_j)
          4: S*_j = LO4(s_j)
        End case
        F(s*_j)=Evaluate(s*_j)
        if (F(s*_j) < F(s_j))
          s_j=s*_j
        if (stopping condition = true)
          Exit
    end for
end for
```

search algorithm (LOi) is selected and the new conformation is evaluated. If this new solution has a better scoring value, then it replaces the calling species. The algorithm finishes when the budged in number of function evaluations is consumed or when other stopping criteria is fullfiled, not obtaining any improvement after a certain number of executions.

The three basic methods implemented into the local search are briefly described in the following:

- ZIP method: this intelligent heuristic analyzes the composition and sequence of the amino acid chain and tries to group the type of amino acid passed as argument. This reasoning facilitates the speed of convergence to the optimum especially at initial levels because it tends to create cores of amino acids of the same type. Through input parameters the type of amino acid to join in the HP model can be specified, i.e. "H" or "P". "H" type is usually the selected as it is intended to group the hydrophobic amino acids. In addition the number of transformations applied to the total substring found can also be defined, when it is less than 100%, the substrings are selected randomly.
- Quake method: this heuristic intentionally modifies the form of a species using an earthquake fault-shaped movement in the chain of amino acids, and it allows variations only in certain portions of the sequence. The parameter *intensity* indicates the number of amino acids that lines up the fault from the epicenter amino acid. The parameter *quantity* represents the number of earthquakes to be applied to the folding sequence.
- Shake method: this heuristic is the one with the highest level of randomness because it only randomly modifies the movements of the folding sequence. The two main parameters are *quantity*, which indicates the number of perturbations to be applied and *intensity*, which represents the number of contiguous positions to be randomly modified from a random amino acid.

And the four recipes or local search procedures are:

- LO1: applies *zip* method to group "H" amino acids in all the string.
- LO2: applies *quake* method with only 1 quake of intensity 2.
- LO3: applies *shake* method with intensity 2 at a only one point of the string.
- LO4: applies two consecutive *shake* operations. The first one with intensity 2 at three different points of the string, and the second one with intensity 2 at a single point.

4 Validation of UEGO using real protein structures

In order to compare obtained solutions by UEGO for the protein folding problem through the HP model with real protein structures, it is necessary to convert coarse-grained HP solutions to all-atom models. For performing such task, the following procedure is carried out:

1. Distance between amino acids is rescaled to 3.8 Å, since this is the typical distance between alpha carbons.
2. Starting from the previous information from coordinates of C-alpha atoms, all-atom reconstruction is performed using the Pulchra package [27].
3. In order to avoid clashes, an energy minimization of the all-atom model is performed using Molecular Operating Environment, MOE, (Chemical Computing Group Inc.) using default parameters and the MMF94 forcefield [13].
4. Crystallographic structural information from the protein is downloaded from the Protein Data Bank (PDB) [5].
5. PyMOL software [8] is used for performing the following tasks:
   (a) Alignment between the obtained all-atom model and the real protein structure.
   (b) Calculation of the degree of spatial similarity among the alpha carbon backbone, the obtained all-atom model and the real protein structure, reported through the root mean square displacement (RMSD).
   (c) Visual representation and inspection of the superposed structures.

The previous methodology has been applied to UEGO solutions in such a way that HP obtained solutions were converted into all-atom models so that comparison with real proteins could be carried out. One representative example can be seen in Fig. 2 for a 13 residue protein where a good agreement is obtained for a small size protein (Tryptophan Zipper 1) and where the RMSD obtained is 3.8 Å.

5 Evaluation of UEGO solutions of the HP protein folding problem

In order to evaluate the performance of the algorithm, a benchmark consisting of eleven 27-monomer sequences that are referenced in [18] has been considered. Since UEGO has a heuristic nature each run may provide a different solution. Thus, to study its robustness, each problem has been solved 100 times and average values and their confidence intervals have been computed. At each run, we obtain the optimal scoring value, the optimal conformation, the number of scoring evaluations employed by the algorithm. With this information we study whether UEGO has successfully found the best known solution [18].

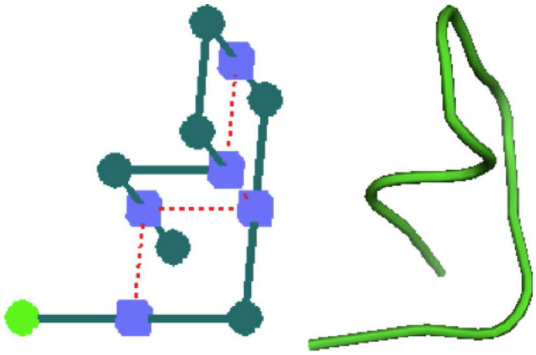

Fig. 2 Conversion of the HP results obtained by UEGO to an all-atom model for Tryptophan Zipper 1. On the *left* we can see the 3D HP model obtained by UEGO, and on the *right* the backbone of the all-atom model after conversion. The RMSD obtained is 3.8 Å

Table 1 Optimum values of the scoring function computed by hELP [18], MC [31], GA [31], mGA [7] and UEGO

HP sequence	hELP	MC	GA	mGA	UEGO
$PHPHPH_3P_2HPHP_{11}H_2P$	−9	−7	−9	−8	−9
$PH_2P_{10}H_2P_2H_2P_2HP_2HPH$	−10	−9	−9	−10	−10
$H_4P_5HP_4H_3P_9H$	−8	−6	−8	NA	−8
$H_3P_2H_4P_3HPHP_2H_2P_2HP_3H_2$	−15	−11	−15	−15	−15
$H_4P_4HPH_2P_3H_2P_{10}$	−8	−7	−8	−8	−8
$HP_6HPH_3P_2H_2P_3HP_4HPH$	−12	−9	−11	NA	−12
$HP_2HPH_2P_3HP_5HPH_2PHPHPH_2$	−13	−10	−12	−13	−13
$HP_{11}HPHP_8HPH_2$	−4	−4	−4	−4	−4
$P_7H_3P_3HPH_2P_3HP_2HP_3$	−7	−6	−7	−7	−7
$P_5H_2PHPHPHPHP_2H_2PH_2PHP_3$	−11	−9	−11	NA	−11
$HP_4H_4P_2HPHPH_3PHP_2H_2P_2H$	−16	NA	NA	NA	−16

The values between parenthesis show the average of Kilo-evaluations of objective function

In addition, UEGO has been compared to other methods referred to in literature, such as hELP (heuristic Energy Landscape Paving) [18], Monte Carlo (MC) [31] and different Genetic Algorithms (GA and mGA) [31]. In Table 1 the scoring value of the global optima for each algorithm is shown. The results confirm that UEGO has a 100% rate of success, i.e., it has always found, in all the runs and in all the problems the best known solution. It can be seen that the best results (reported by the value of the scoring function in Table 1) are obtained by both hELP and UEGO, obtaining the same lower minimum values. In contrast, MC and GA methods do not always obtain these lower minimum values and mGA does not yield solutions in half of the cases.

Regarding computational efficiency of the different methods, results showed that UEGO outperforms MC and genetic algorithms in terms of number of scoring evaluations, though it requires more evaluations than hELP.

6 Conclusions and future works

In this work we have applied the methodology of the UEGO, to solve the protein structure optimization problem based on the HP model. In our study, the HP protein folding solutions have been defined as means of conformations encoded by relative coordinates, and a new and efficient local optimizer has been designed based on the characteristics of the protein model.

When comparing with results from literature it can be seen that UEGO outperforms Monte Carlo and Genetic Algorithms, and obtains a similar performance to the hELP method. Furthermore, UEGO can find all the global and local optima due to its multimodality.

An additional advantage of our methodology is that apart from being compared to synthetic benchmarks, we have shown that our 3D HP predictions can be converted into all-atom models and predict with a reasonable accuracy the structure of small size proteins.

As future work we plan to improve the efficiency of the UEGO algorithm by means of: (1) improving the local optimizer algorithm by incorporating the keys of the hELP method; (2) taking advantage of the UEGO intrinsic parallelism on High Performance Computing platforms.

Acknowledgments This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2012-37483-C03-03), Junta de Andalucía (P10-TIC-6002 and P12-TIC301), Fundación Séneca (The Agency of Science and Technology of the Region of Murcia, 00003/CS/10, 15254/PI/10 and 18946/JLI/13) and by the Nils Coordinated Mobility under Grant 012-ABEL-CM-2014A, in part financed by the European Regional Development Fund (ERDF).

References

1. A. Arkhangel'skii, *General Topology I: Basic Concepts and Constructions Dimension Theory (Encyclopaedia of Mathematical Sciences)* (Springer, Berlin, 2011)
2. A. Arrondo, J. Fernández, J. Redondo, P. Ortigosa, An approach for solving competitive location problems with variable demand using multicore systems. Optim. Lett. **8**, 555–567 (2013)
3. R. Backofen, S. Will, A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. Constraints **11**(1), 5–30 (2006)
4. B. Berger, T. Leighton, Protein folding in the hydrophobic–hydrophilic (hp) model is np-complete. J. Comput. Biol. **5**(1), 27–40 (1998). doi:10.1089/cmb.1998.5.27
5. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000)
6. T.N. Bui, G. Sundarraj, An efficient genetic algorithm for predicting protein tertiary structures in the 2D hp model, in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, GECCO '05 ACM*, (New York, NY, USA, 2005), pp. 385–392
7. F. Custodio, H. Barbosa, L. Dardenne, Investigation of the threedimensional lattice HP protein folding model using a genetic algorithm. Genet. Mol. Biol. **27**(4), 611–615 (2004)
8. W.L. DeLano, *The PyMOL Molecular Graphics System* (DeLano Scientific LLC, San Carlos, CA, 2002)
9. K.A. Dill, S. Bromberg, K. Yue, H.S. Chan, K.M. Ftebig, D.P. Yee, P.D. Thomas, Principles of protein folding a perspective from simple exact models. Protein Sci. **4**(4), 561–602 (1995)
10. K.A. Dill, J.L. MacCallum, The protein-folding problem, 50 years on. Science **338**(6110), 1042–1046 (2012)
11. S. Fidanova, I. Lirkov, Ant colony system approach for protein folding. in *IMCSIT*, pp. 887–891. IEEE (2008)

12. J. García-Martínez, E. Garzón, P. Ortigosa, A GPU implementation of a hybrid evolutionary algorithm: GPuEGO. J. Supercomput. 1–12 (2014). doi:10.1007/s11227-014-1136-7
13. T.A. Halgren, Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. J. Comput. Chem. **17**(5–6), 490–519 (1996)
14. M. Jelásity, P. Ortigosa, I. García, UEGO, an abstract clustering technique for multimodal global optimization. J. Heuristics **7**(3), 215–233 (2001)
15. F. Khatib, S. Cooper, D. Tyka, All: Algorithm discovery by protein folding game players. PNAS 108 (47) (2011). http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3223433/pdf/pnas.1115898108
16. I. Kondov, Protein structure prediction using distributed parallel particle swarm optimization. Nat. Comput. **12**(1), 29–41 (2013)
17. C. Lin, S. Su, Protein 3D hp model folding simulation using a hybrid of genetic algorithm and particle swarm optimization. Int. J. Fuzzy Syst. **13**(2), 140–147 (2011)
18. J. Liu, G. Li, J. Yu, Y. Yao, Heuristic energy landscape paving for protein folding problem in the three-dimensional HP lattice model. Comput. Biol. Chem. **38**, 17–26 (2012)
19. P. Ortigosa, I. García, M. Jelásity, Reliability and performance of UEGO, a clustering-based global optimizer. J. Global Optim. **19**(3), 265–289 (2001)
20. P. Ortigosa, J. Redondo, I. García, J. Fernández, A population global optimization algorithm to solve the image alignment problem in electron crystallography. J. Global Optim. **37**(4), 527–539 (2007)
21. S. Ozkan, G.A. W., J. Chodera, K. Dill, Protein folding by zipping and assembly. PNAS 104 (29) (2007). http://www.pnas.org/content/104/29/11987.full+html?with-ds=yes
22. J. Peña, J. Cecilia, H. Pérez-Sánchez, Application of ant colony optimization in a hybrid coarse-grained and all-atom based protein structure prediction strategy. in *13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2013*, pp. 1154–1156 (2013)
23. J. Redondo, J. Fernández, A. Arrondo, I. García, P. Ortigosa, Fixed or variable demand? Does it matter when locating a facility? Omega **40**(1), 9–20 (2012)
24. J. Redondo, J. Fernández, A. Arrondo, I. García, P. Ortigosa, A two-level evolutionary algorithm for solving the facility location and design (1|1)-centroid problem on the plane with variable demand. J. Global Optim. **56**(3), 983–1005 (2013)
25. J. Redondo, J. Fernández, I. García, P. Ortigosa, Parallel algorithms for continuous competitive location problems. Optim. Methods Softw. **23**(5), 779–791 (2008)
26. J. Redondo, J. Fernández, I. García, P. Ortigosa, A robust and efficient global optimization algorithm for planar competitive location problems. Ann. Oper. Res. **167**, 87–106 (2009)
27. P. Rotkiewicz, J. Skolnick, Fast procedure for reconstruction of full-atom protein models from reduced representations. J. Comput. Chem. **29**(9), 1460–1465 (2008)
28. A. Schug, W. Wenzel, An evolutionary strategy for all-atom folding of the 60-amino-acid bacterial ribosomal protein L20. Biophys. J. **90**(12), 4273–4280 (2006)
29. A. Shmygelska, R. Aguirre-Hernndez, H.H. Hoos, An ant colony optimization algorithm for the 2D hp protein folding problem. in *Proceedings of the 16th Canadian Conference Artificial Intelligence*, (Springer, 2002), pp. 400–417
30. T. Strunk, M. Wolf, W. Wenzel, Peptide structure prediction using distributed volunteer computing networks. J. Math. Chem. **50**(2), 421–428 (2012)
31. R., Unger, J. Moult, A genetic algorithm for 3D protein folding simulations. in *Proceedings of the Fifth Annual International Conference on Genetic Algorithms*, (Kaufmann, San Francisco, 1993) , p. 581–588
32. R. Unger, J. Moult, Genetic algorithms for protein folding simulations. J. Mol. Biol. **231**(1), 75–81 (1993)