J Math Model Algor (2010) 9:181–194
DOI 10.1007/s10852-010-9125-1

Protein Structure Prediction Using Bee Colony Optimization Metaheuristic

Rasmus Fonseca · Martin Paluszewski · Pawel Winter

Received: 1 March 2009 / Accepted: 1 December 2009 / Published online: 19 February 2010

Abstract Predicting the native structure of proteins is one of the most challenging problems in molecular biology. The goal is to determine the three-dimensional structure from the one-dimensional amino acid sequence. *De novo* prediction algorithms seek to do this by developing a representation of the proteins structure, an energy potential and some optimization algorithm that finds the structure with minimal energy. Bee Colony Optimization (*BCO*) is a relatively new approach to solving optimization problems based on the foraging behaviour of bees. Several variants of BCO have been suggested in the literature. We have devised a new variant that unifies the existing and is much more flexible with respect to replacing the various elements of the BCO. In particular, this applies to the choice of the local search as well as the method for generating scout locations and performing the waggle dance. We apply our BCO method to generate good solutions to the protein structure prediction problem. The results show that BCO generally finds better solutions than simulated annealing which so far has been the metaheuristic of choice for this problem.

Keywords Protein structure prediction · Bee Colony Optimization · Metaheuristic

Partially supported by a grant from the Danish Research Counsil (51-00-0336).

R. Fonseca (✉) · P. Winter
Department of Computer Science, University of Copenhagen, Copenhagen, Denmark
e-mail: rfonseca@diku.dk

P. Winter
e-mail: pawel@diku.dk

M. Paluszewski
The Bioinformatics Centre, University of Copenhagen, Copenhagen, Denmark
e-mail: palu@binf.ku.dk

1 Introduction

Proteins are the primary building blocks in all living organisms. They are made of amino acids bound together by peptide bonds. Depending on the sequence of amino acids, the proteins fold in three dimensions so that the Gibbs energy is minimized. The shape determines the function of the protein. *Protein structure prediction* (PSP) is the problem of predicting this three-dimensional structure from the amino acid sequence and is considered one of the most important open problems of theoretical molecular biology. The PSP problem has applications in medicine within areas like drug- and enzyme design [12].

PSP proves to be a very difficult optimization problem. Solving it exactly is only possible when using very simplified models. Use of heuristics is therefore necessary for more detailed models and energy functions. However, even in simplified scenarios, many computational problems arise. One of these problems is the belief that free energy landscapes tend to have many local minima [11].

Lately, several optimization heuristics inspired by bee colonies have been proposed. The two main approaches are the evolutionary algorithms and the foraging algorithms. The evolutionary approach was initially proposed in [1] and was based on the mating of bee drones with a queen bee. The foraging approach was proposed simultaneously in [17, 18] and [8, 9] and mimics the foraging behaviour of honey bees searching for and collecting nectar in a flower field. This heuristic, like real honey-bees, performs a wide search for good solutions and has a flexible method for allocating resources to intensify the local searches. This seems like a good strategy in the PSP to avoid getting stuck in the local minima of the energy landscape. Several names have been given to the foraging algorithm but here *Bee Colony Optimization* (BCO) is chosen.

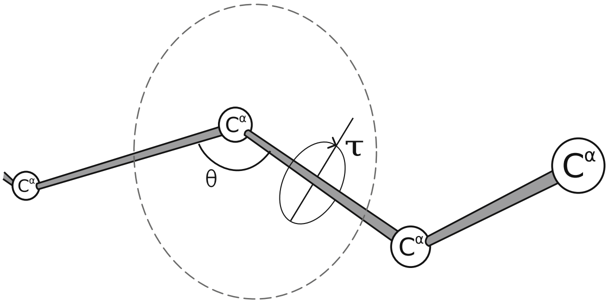
Bahamish et al. [2] previously used the *Bees Algorithm* [17] to find the native state of the 5-residue peptide ’met-enkalphin’ (PDB-ID: 1PLW) using a full resolution torsion angle-based representation. We apply the BCO metaheuristic to the PSP problem for real-sized proteins using a simplified representation. Good quality solutions, often called decoys, in terms of the RMSD and GDT similarity measures, are generated. These decoys can be used as starting solutions for more advanced methods. Since a coarser representation is used, real-sized protein structures can be attacked by the BCO metaheuristic. This is the first time a bee heuristic has been used to predict the structure of real-sized proteins (more than 50 residues). We do not claim to solve the PSP or even compete with state-of-the-art PSP algorithms like Rosetta [19] or I-Tasser [27]. However, the BCO metaheuristic has appealing properties. For instance, the scout bees make sure several local minima are searched, and the waggle dances intensify the search in promising areas. We believe this makes BCO suitable for the PSP.

In Section 2 the representation and the energy function of proteins are described. In Section 3 our adaptation of BCO is specified. Finally, experiments are described in Section 4 and discussed in Section 5.

2 Protein Model

The representation of proteins is important since it determines the size and conformation of the search-space. The following section describes our representation of the proteins structure.

Fig. 1 C_α trace of backbone. Each amino acid is assigned two angles: θ and τ



2.1 Segment Representation

There are 20 different kinds of amino acids, each represented by a letter. The letter-sequence of amino acids is called the *primary structure* of the protein. Frequently occurring local structures of amino acids, such as helices and strands, are called *secondary structures* and the full description of the protein (i.e., 3D coordinates of all atoms) is called the *tertiary structure*.

When trying to determine the overall tertiary structure of a protein, sometimes the side chains and the atoms of the backbone are disregarded, and only the central carbon atom, C_α, of amino acids are represented. This leads to the C_α-trace representation of proteins illustrated in Fig. 1. The entire protein structure can be represented by assigning two angles to each amino acid, θ and τ.

Each amino acid of a protein can be classified as belonging to exactly one secondary structure. Here three classes of secondary structures are considered: helix, strand and coil. Helices and strands are distinguished by the unique geometrical layout of the $C_\alpha$atoms in the tertiary structure (left part of Fig. 2) which is caused by a special pattern of hydrogen bonds. Coil is the class of amino acids that do not have a regular pattern of hydrogen bonds, and therefore has only few geometric constraints on the tertiary structure.

A sequence of C_α-atoms of the same secondary structure class is here called a *segment*. The secondary structure class of all amino acids is predicted and used as part

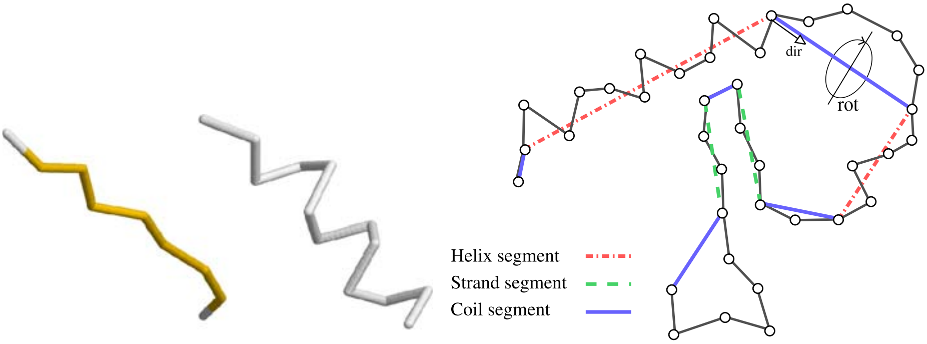


Fig. 2 *Left*: typical structures for strand and helix (generated using RasMol [20]). *Right*: segment representation of a protein

of our model. Segments can be considered as rigid rods that define the overall path of C_α-atoms belonging to the segment. Segments always have a start coordinate and a direction, and for helices and strands their end-coordinate can also be determined because of their constrained geometry. A segment is an abstract representation and does not explicitly contain the coordinates of internal C_α-atoms. The specification of how the C_α-atoms are arranged around the segment is called the *segment structure*. Together, all segments and segment structures constitute the *complete structure*, or simply the protein structure. The right part of Fig. 2 is an illustration of a complete structure using the segment representation.

The tertiary structure of any protein can be described by a complete structure. However, to discretize and reduce the conformational space of this model, the degree of freedom for segments and segment structures is reduced. Segments are therefore only allowed to have a discrete number d of predefined directions between the first and last C_α-atoms. Also, the number of possible segment structures is limited to s. The method used to determine the segment structures of helix, strand and coil-classes is described in Section 2.2. Obviously, the chance of being able to represent a complete structure similar to the native structure of the protein increases when more directions and segment structures are allowed, but this also increases the size of the conformational space.

The complete structure of a protein with m segments is represented by a pair of integers for each segment indicating the segment direction and segment structure.

$$(d_i, s_i), \qquad i = 1 \dots m, \quad d_i \in \{1 \dots d\}, \quad s_i \in \{1 \dots s\}$$

2.2 Segment Structures

In this section it is described how the s allowed segment structures of a given segment are computed. This computation depends on the secondary structure class of the segment.

Helix and Strand Structures The most observed angle pair for an amino acid is $(\theta, \tau) = (91°, 49°)$ in helices and $(\theta, \tau) = (120°, 163°)$ in strands. Given a helix or strand segment, one segment structure having these angle properties is generated. Then the other $s - 1$ segment structures are generated by rotating the first structure uniformly around the axis going through the first and last C_α-atoms.

Coil Structures There are no simple geometric constraints that describe coil structures. However, experiments show that short sequences with similar amino acid sequences, so-called homologous sequences, often have similar tertiary structures [4]. Given a coil segment, the PDBSelect-25 dataset [3] is queried to find the $\sqrt{s}$ best matching structures. Each of these structures is rotated uniformly $\sqrt{s}$ times such that a total of s segment structures is obtained.

2.3 Energy

Determining a simple energy function for protein structures that is computationally fast and correlates well to native structures is still an open problem. Pseudo-energy functions are based on statistical analysis of large sets of proteins. These types of energy functions are usually very fast but the quality of the minimal energy structures varies greatly.

A promising pseudo-energy function described in [16] is based on *Half-Sphere Exposure* (HSE) [5] and *Contact Numbers* (CN). This function requires very little computation and represents many of the crucial aspects of native structures (for instance side-chain surface exposure and residue burial). An important property of the HSE and CN measures is that they can both be predicted fairly accurately, so the energy of a structure can be calculated as the deviation from the predicted values.

For a given amino acid, the HSE is a pair of integers describing how many amino acids are contained in a half-sphere *above* the amino acid and how many are contained in the half-sphere *below* (See Fig. 3). The plane dividing the two half-spheres is specified by the position of the C$_\alpha$-atom, A_i, and an $\overrightarrow{up}$-vector specific to the amino acid. The $\overrightarrow{up}$-vector can be defined in the following way.

$$\overrightarrow{up} = \overrightarrow{A_{i-1}A_i} + \overrightarrow{A_{i+1}A_i}$$

This $\overrightarrow{up}$ vector is undefined for the first and last amino of the protein, so for these only the contact number *CN* can be calculated. The CN for every amino acid is the number of amino acids contained in the *entire* sphere. The HSE and CN vectors specifying all the up/down numbers and contact number can be predicted from the primary structure alone using support vector regression [23, 25].

Let $\mathcal{P}$ denote the conformational space of a protein with n amino acids. Let $p \in \mathcal{P}$. The total energy $Q(p)$ is defined as the sum of the individual energy contributions $Q_p(i)$ from each amino acid i, i.e.,

$$Q(p) = \sum_{i=1}^{n} Q_p(i) \tag{1}$$

where

$$Q_p(i) = \begin{cases} \Delta CN(i)^2 & \text{if } i \text{ is the first amino acid of a segment.}^1 \\ \Delta HD(i)^2 + \Delta HU(i)^2 & \text{otherwise} \end{cases}$$

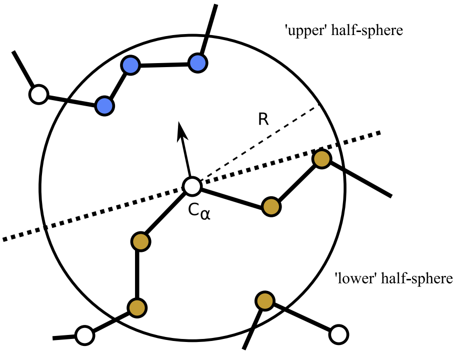


Fig. 3 Half-sphere exposure for an amino acid. The up/down pair is (3, 5). The contact number is 8

and

- $\Delta CN(i)$ is the difference between the actual contact number of the i-th amino acid and the desired (i.e., predicted).
- $\Delta HD(i)$ is the difference between the down half sphere exposure number of the i-th amino acid and the desired down half sphere exposure number.
- $\Delta HU(i)$ is the similar difference for the up half sphere exposure.

A radius of the HSE-sphere around 13 Å is known to give a good prediction quality [24] and it seems to capture both local and non-local contacts. The optimal radius has yet to be determined, both in terms of predictability and information content.

Since many amino acids are hydrophobic, globular proteins fold into tight spheric conformations. An HSE based energy function is not enough to ensure this behaviour, so the mean squared distance of the amino acids from the protein center, the *radius of gyration* (Rg), is introduced. The center is defined as the average position of C_α atoms in the structure. Rg can be predicted from the number of amino acids n of the protein [22]:

$$Rg_{\text{pred}} = 2.2n^{0.38} \tag{2}$$

This prediction is often accurate for globular proteins. Infinite energy is therefore assigned to structures having $Rg > 1.2 \cdot Rg_{\text{pred}}$.

A structure is said to be clashing if the distance between two C_α-atoms is less than 3.5 Å. A clashing structure is also assigned infinite energy.

3 Bee Colony Optimization

In nature, a foraging bee can be said to be in one of three states: A scout bee, a worker bee or an onlooker. Scout bees fly around a flower field at random and when a flowerbed is found, they return to the hive and perform a waggle dance. The dance indicates the estimated amount of nectar, direction and distance to the flowerbed. Onlooker-bees present in the hive watch different waggle dances, choose one and fly to the selected flowerbeds to collect nectar. Worker bees act like scout bees except that when they have performed the waggle dance they return to their old flowerbed to retrieve more nectar instead of flying out at random. A bee usually chooses to become a worker bee when the chosen flowerbed has a very high concentration of nectar.

In our adaptation of the BCO metaheuristic, each bee corresponds to a specific complete solution, and the nectar amount corresponds to an objective value in the energy landscape. Sending out scout bees corresponds to finding a random feasible solution and sending out onlookers corresponds to performing a local search iteration on some existing solution. The onlookers choose a solution for local search based on the objective value of scout and worker-solutions in previous iterations.

[1] The reason why CN instead of HSE is used for the first amino acid of each segment is that it was necessary for the Branch and Bound algorithm described in [15, 16]. In order to compare solutions found here with those in [16] the same energy function is preserved.

This method is largely the *Bees Algorithm* proposed in [18]. In a non-changing solution space, the fitness of a solution does not deplete in the same way a real life flowerbed depletes of nectar. Exhaustion is therefore forced when a worker-solution can not be improved. This idea is somewhat similar to the idea of pruning parts of the search space as described in [14]. The process of exhausting a search around a worker-solution is proposed as part of the *Artificial Bee Colony* algorithm described in [9]. Our adaptation of the BCO metaheuristic is a synthesis of these two approaches.

Algorithm 1 BCO pseudocode

```
$saved \leftarrow \emptyset$
$pop \leftarrow \textsc{ScoutStrategy}(W + S)$
while Stopping criterion is not met
    for each $p \in pop$ do
        $onlookers[p] \leftarrow \textsc{OnlookerStrategy}\left(\textsc{Cost}(p), \sum_{p'} \textsc{Cost}(p'), O\right)$
        $p \leftarrow \textsc{NeighborhoodSearch}(p, onlookers[p])$
        if $\textsc{Cost}(p)$ has not improved for $Exhaust$ iterations then
            $saved \leftarrow saved \cup \{p\}$
            $p \leftarrow \textsc{ScoutStrategy}(1)$
    $newScouts \leftarrow \textsc{ScoutStrategy}(S)$
    Replace the $S$ solutions in $pop$ that has the worst costs with $newScouts$
return The best solution—either from $pop$ or from $saved$
```

Here S, W and O is the number of scout, worker and onlooker bees, respectively. OnlookerStrategy is the strategy for assigning onlookers and NeighborhoodSearch(p, o) is the neighborhood strategy for performing o iterations of local search around a solution, p. ScoutStrategy(s) is a method for generating a set of s random solutions. The S worst solutions in the population are always scout-solutions which means that the W best ones are worker-solutions. It is not specifically indicated how new solutions should be generated using ScoutStrategy, how onlookers should be assigned or how local search in NeighborhoodSearch should be performed. Depending on the nature of a problem each of these three methods can be designed to fit the problem. New solutions can be generated with genetic algorithms by using mutation and crossover in ScoutStrategy, and any of the numerous existing local search heuristics can be used as NeighborhoodSearch. The basic procedure however is to let ScoutStrategy generate a random solution and set NeighborhoodSearch to perform hill-climbing. Using this basic procedure it is observed that:

$$\mathrm{BCO}(S, W, O, Exhaust = \infty) = \mathrm{BA}(S, W, O)$$

$$\mathrm{BCO}(S = 0, W, O, Exhaust) = \mathrm{ABC}(W, O, Exhaust)$$

Where BA is the Bees Algorithm [18] and ABC the Artificial Bee Colony algorithm [8]. The above representation of the foraging bees optimization paradigm is more generally applicable than the ones presented in [18] and [8] since both are special instances of BCO.

3.1 Bee Colony Optimization Applied to PSP

Algorithm 1 can be used for any optimization problem where OnlookerStrategy, NeighborhoodSearch and ScoutStrategy are defined, so to utilize BCO for PSP these three methods have to be specified. The energy function $Q(p)$ (Eq. 1) is used as cost-function.

*Scout Search Strategy (*ScoutStrategy*)* To find a random feasible solution, a depth first search is used to determine the direction d_i and structure s_i of each segment i. At each level in the depth first search, a random ordering of direction and structure is tried so the same solution is not generated every time.

*Onlooker Choosing Strategy (*OnlookerStrategy*)* A number of onlookers are assigned to a solution j among the scouts and workers based on the costs of the population and the amount of onlookers O. Onlookers are assigned probabilistically based on a fitness given by:

$$fitness(j) = \frac{1/\textsc{Cost}(j)}{\sum_{j'} 1/\textsc{Cost}(j')}$$

The OnlookerStrategy, however ensures that only a total of O onlookers are assigned in each iteration.

*Onlooker Neighborhood Strategy (*NeighborhoodSearch*)* Any local search could be utilized as neighborhood strategy but a simple hill-climbing strategy is chosen. A neighbor solution is generated by changing directions d_i of two randomly chosen segments and the segment structure s_i of four randomly chosen segments. If the cost improves the new solution is accepted.

4 Experiments and Results

Two sets of experiments are performed, one on a simple model and one on a flexible model.

The *simple model* allows $d = 12$ basic directions defined by the face-centered lattice and $s = 8$ rotations for each segment: $d_i \in \{1..12\}, s_i \in \{1..8\}$. This choice of d and s ensures a reasonable flexibility of the structure, but also makes the search-space sufficiently small to allow the EBBA algorithm [16] to find an optimal solution within 48 h.

The *flexible model* allows $d = 73$ basic directions defined by a combination of the face-centered, body-centered and simple cubic lattices and $s = 16$ rotations for each segment: $d_i \in \{1..73\}, s_i \in \{1..16\}$. This choice of d and s is made to ensure that the model is much more flexible than the simple model. The purpose of creating a flexible model is to see if a metaheuristic can find solutions with lower energy than the optimal energy found in the simple model.

The energy function, $Q(p)$, is the same for both models. The distance measure RMSD(p) and the Global Distance Test (GDT) measure [26] can be compared for structures belonging to both of these models as well as structures obtained using completely different models and methods. GDT$_c(p)$ is calculated as the largest set of amino acids in some structure p that can be superposed on to the native structure such that the distance of each amino acid in the set is less than c from the amino acid position in the native structure. $\mathrm{GDT}(p)$ is defined as the average of $\mathrm{GDT}_1(p)$, $\mathrm{GDT}_2(p)$, $\mathrm{GDT}_4(p)$ and $\mathrm{GDT}_8(p)$.

The first set of experiments uses BCO, EBBA and a simulated annealing algorithm to find good decoys using the simple model for six proteins. These six proteins have previously been used for benchmarks in the literature [6, 16, 21]. The input to the optimization algorithms is a secondary structure assignment, the HSE-vectors and the radius of gyration. For each protein these values are obtained using prediction tools. Based on the amino acid sequence, the secondary structure is predicted using PSIPRED [13] and HSE-vectors using LAKI [24]. For better comparison of energy levels, the HSE predictions from [16], which were done using LAKI [24], were used. The radius of gyration is predicted using Eq. 2. The six benchmark proteins used here also exist in PDB, so there is a slight chance that the training sets for PSIPRED and LAKI contain some of these proteins. However, the prediction quality of the six benchmark proteins is close to what should be expected. We therefore do not consider it to be a problem that the benchmark proteins exist in PDB.

In order to be able to compare the results from BCO with the results obtained using EBBA [16], BCO is run for 48 h. Due to this large run-time extensive parameter-optimization has not been considered. Considering the runtimes for generating a scout and finding a valid neighbor-solution, we estimate that using $S = 10$ scouts, $W = 10$ workers, $O = 100$ onlookers and $Exhaust = 5$ is appropriate. We also wish to compare BCO to a commonly used metaheuristic within bioinformatics, so a Simulated Annealing (SA) algorithm is implemented. New solutions are generated using the ScoutStrategy method described in Section *Scout Search Strategy (ScoutStrategy)* and the local search is based on the NeighborhoodSearch method also described in Section *Onlooker Neighborhood Strategy (NeighborhoodSearch)*. The only difference in the NeighborhoodSearch method is that the SA algorithm accepts solutions that have ΔQ worse energy with the probability $p = e^{\frac{-\Delta Q}{T}}$. We tested different starting temperatures from 0.5 to 16 and measured the ratio of accepted changes out of all the valid changes. Johnson et al. [7] studied the simulated annealing heuristic and found that this ratio should be between 20% and 90%. We therefore set the starting temperature to $T = 1$ because the resulting acceptance ratio was around 50%. A linear annealing schedule was chosen such that the temperature falls to $T = 0$ when SA terminates. This eliminates the need to decide a final temperature.

Ideally SA converges on optimal solutions if allowed to cool down sufficiently slow. To improve the quality of solutions obtained by SA, restarts are often suggested in the literature. We settled on a relatively low number of restarts (ten restarts, each taking 4.8 h) as more restarts would make SA a special variant of BCO.

Figure 4 shows a single run of EBBA and five runs of BCO. The time-cost curves show that BCO is very stable and finds the optimal value of $Q(p)$ almost as fast as EBBA. Similar experiments were performed for SA but it always converged on a suboptimal value after roughly 5–6 h.

The second round of experiments are performed using BCO and SA, but this time on the flexible model. Using the flexible model EBBA can no longer find the optimal value in 48 h. In addition to the six previously benchmarked proteins we also attempt to predict good structures for two somewhat bigger targets from

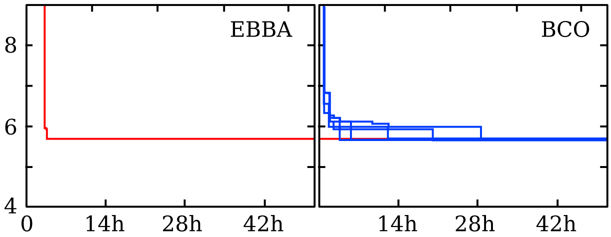


Fig. 4 Best observed energy vs. time for EBBA and BCO using the simple model

CASP7 [10]. We have intentionally chosen a pair that proved to be hard to predict by CASP7 participants. Most successful CASP7 methods were homology-based. Since our algorithm is not using homology modelling, it should be compared with PSP methods for proteins with no good templates in PDB. For the two CASP proteins, the newer and more accurate HSE prediction server HSEpred [23] was used instead of LAKI.

To make BCO and SA collect decoys, we stored the 1,000 best structures with respect to $Q(p)$ encountered during a search. For comparison and evaluation of the model and prediction quality, the second round of experiments was also done using the exact secondary structures, exact HSE-vectors and exact radius of gyration

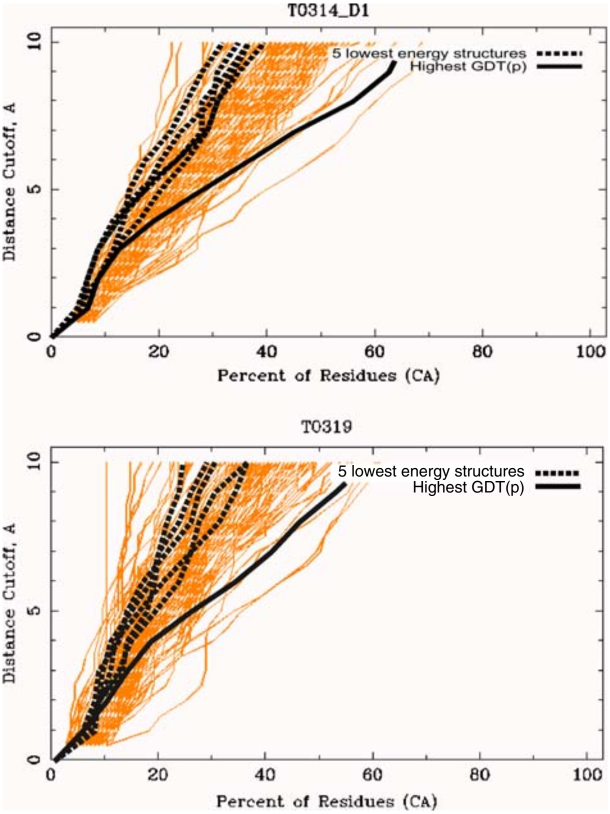


Fig. 5 GDT analysis plot for the proteins 2HG6 (*top*) and 2J6A (*bottom*). The x-axis shows the offset, *c*, and the y-axis shows $\mathrm{GDT}_c(p)$. The *orange curves* are structures from all the predictors at CASP7. The *blue curves* are the five structures with lowest energy $Q(p)$ generated by BCO. The *green curve* is the structure $p^\dagger$ with highest $\mathrm{GDT}(p)$. *Both blue and green curves* are found using predictions of secondary structure, HSE vectors and radius of gyration

obtained from the native structures of the proteins. These structures cannot be considered solved *de novo*. All computations were performed on a 3.4 GHz Intel Xeon with 2 GB RAM.

Table 1 summarizes the results of the runs from BCO and SA using the flexible model, EBBA using the simple model as well as the results from CASP7. p^* is the protein structure encountered during a search for which the energy $Q(p)$ is lowest. For BCO, SA and EBBA the energy function is identical. $p^\dagger$ is the protein structure—among the 1,000 saved decoys—for which the similarity $(\mathrm{GDT}(p))$ is highest.

Figure 5 show GDT analysis plots for the proteins 2HG6 and 2J6A. The GDT analysis is a type of plot used at CASP to indicate how good the $\mathrm{GDT}_c(p)$ is, using different distance cutoff values c. A curve lying to the far right correspond to a conformation near the native structure. All the orange curves in the figures correspond to structures generated by the participants at CASP7. The blue curves all correspond to structures generated by BCO using predicted secondary structure, predicted HSE-vectors and predicted radius of gyration. The green curves correspond to $p^\dagger$ structures with highest $\mathrm{GDT}(p)$.

5 Discussion and Conclusion

The results of BCO and SA compared to those achieved at CASP7 are shown for the proteins 2HG6 and 2J6A in Table 1. It can be seen that the HSE-based energy function does not completely identify the best structure since $\mathrm{GDT}(p^*)$ is relatively low for BCO and SA. This can also be observed from the GDT analysis in Fig. 5. The five structures with lowest energy result in curves that are slightly worse than the average at CASP7. If, however, a more advanced energy function is applied to the 1,000 generated structures then $p^\dagger$ can possibly be identified. Using $\mathrm{GDT}(p)$ as quality measure this would rank the structures obtained by BCO as 30-th out of 132 for 2HG6 and 17-th of 132 for 2J6A at CASP7. The curves illustrating $p^\dagger$ in the GDT analysis are even more promising as $\mathrm{GDT}_{10}(p^\dagger)$ are among the highest for both 2HG6 and 2J6A. This indicates that the model and the BCO heuristic are good at finding structures where the ’overall’ conformation is close to the native. This is a notable achievement for an energy function that is primarily based on predicted HSE numbers and radius of gyration.

When comparing BCO to SA, the focus should be on the values of $Q(p^*)$ since both algorithms optimize the energy. For all the problems, except 2GB1 exact, BCO achieves a lower value of $Q(p^*)$ which indicates that BCO is superior to SA on these types of problems. The average values of $Q(p^*)$ for the six smaller proteins are illustrated in Table 2. For these proteins BCO finds values of $Q(p^*)$ that, on average, are 5% better than those found by SA. It is worth noting that Monte-Carlo based algorithms like SA usually are the metaheuristics of choice for PSP.

When looking at the results for 1FC2 (exact) and 1ENH (exact), it is clear that they differ from the other rows. The lowest energy observed is less than 3 for both runs which is considerably lower than for the other runs. It is remarkable that the corresponding very low energy structures are native-like. This supports the hypothesis that HSE, secondary structure and radius of gyration contain enough information to identify the native structure of the protein. There are two possible

Table 1 Results from Bee Colony Optimization (BCO), Simulated Annealing (SA), Efficient Branch and Bound Algorithm (EBBA) and CASP7

PDB	Size	SS &	BCO				SA			EBBA		CASP7
id		energy	$Q(p^*)$	RMSD(p^*)	GDT(p^*) (%)	GDT$(p^\dagger)$ (%)	$Q(p^*)$	GDT(p^*) (%)	GDT$(p^\dagger)$ (%)	$Q(p^*)$	RMSD(p^*)	GDT(p^*)
1FC2	43	Pred.	3.65	6.62	52.33	55.23	3.76	47.67	58.14	5.26	8.4	–
		Exact	1.94	1.65	83.72	84.30	2.62	66.28	79.07	4.34	6.6	–
1ENH	54	Pred.	4.67	6.99	40.28	50.93	4.91	40.28	50.46	5.70	10.2	–
		Exact	2.91	2.28	71.30	73.61	3.56	54.63	67.13	4.36	3.5	–
2GB1	56	Pred.	5.41	8.86	30.80	41.96	5.50	29.46	42.41	6.22	7.8	–
		Exact	5.52	9.18	31.70	47.32	5.03	27.68	49.11	4.22	4.3	–
2CRO	65	Pred.	3.85	8.76	31.15	42.31	4.44	35.38	39.62	5.89	9.4	–
		Exact	6.10	7.61	35.38	47.69	6.13	41.54	51.54	6.49	9.2	–
1CTF	68	Pred.	5.43	9.01	36.03	38.97	5.74	33.46	37.87	5.84	11.3	–
		Exact	5.67	7.50	38.60	44.12	5.83	25.74	49.63	7.19	11.0	–
4ICB	76	Pred.	4.77	9.02	32.57	38.49	5.32	29.28	44.08	6.79	6.4	–
		Exact	5.38	10.38	28.29	44.41	5.45	28.95	42.11	6.18	7.4	–
2HG6	106	Pred.	6.14	16.26	14.89	22.17	6.61	17.69	27.59	–	–	30.34%
		Exact	4.70	14.49	20.05	24.29	5.19	19.81	30.19	–	–	–
2J6A	136	Pred.	6.79	14.34	14.34	19.30	6.79	17.10	20.59	–	–	27.78%
		Exact	6.20	16.31	18.38	22.98	7.25	17.46	21.88	–	–	–

At CASP7 the proteins 2HG6 and 2J6A had target numbers T0314 and T0319 respectively. Large values of GDT are preferable whereas low values of RMSD are preferred. Since structure prediction seeks to minimize the energy, $Q(p)$ should be as low as possible. p^* is the structure, encountered during search, with lowest energy and $p^\dagger$ is the one with highest GDT. The same combinatorial protein representation is used for BCO and SA. An identical representation is used for EBBA but some parameters diverge

Table 2 Comparison of best energy values for BCO, SA and EBBA when run on 1FC2, 1ENH, 2GB1, 2CRO, 1CTF and 4ICB

	BCO	SA	EBBA
Average $Q(p^*)$	4.61	4.86	5.71
Improvement over EBBA	24%	17%	–
Improvement over SA	5%	–	–

Note that some parameters diverge in EBBA's representation of the protein and EBBA is the only algorithm that guarantees a globally optimal p^*

reasons why we do not find these very low energy structures for the other proteins. One reason could be that native-like structures cannot be represented accurately enough in our model when trying to represent large proteins. The other possibility is that our search algorithm requires more time to find the native-like structure. This is a subject for further investigation. We did perform a fair amount of ad-hoc experiments adjusting the parameters of the search-methods but no results indicated that any parameters were better suited for large proteins than for small.

References

1. Abbass, H.A.: MBO: marriage in honey bees optimization—a haplometrosis polygynous swarming approach. In: Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, pp. 207–214 (2001)
2. Bahamish, H.A.A., Abdullah, R., Salam, R.A.: Protein conformational search using Bees Algorithm. In: Asia International Conference on Modelling and Simulation, pp. 911–916 (2008)
3. Boberg, J., Salakoski, T., Vihinen, M.: Selection of a representative set of structures from Brookhaven protein data bank. Proteins **14**(2), 265–76 (1992)
4. Chothia, C., Lesk, A.M.: The relation between the divergence of sequence and structure in proteins. EMBO J. **5**, 823–826 (1986)
5. Hamelryck, T.: An amino acid has two sides: a new 2D measure provides a different view of solvent exposure. J. Proteins Struct. Funct. Bioinf. **59**(1), 38–48 (2005)
6. Hamelryck, T., Kent, J.T., Krogh, A.: Sampling realistic protein conformations using local structural bias. PLOS Computat. Biol. **2**, e131 (2006)
7. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning. Oper. Res. **37**(6), 865–892 (1989)
8. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes Univ., Engineering Faculty, Computer Engineering Department (2005)
9. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
10. Kryshtafovych, A., Fidelis, K., Moult, J.: Progress from CASP6 to CASP7. Proteins Struct. Funct. Bioinf. **69**(S8), 194–207 (2007)
11. Li, Z., Scheraga, H.A.: Monte Carlo-minimization approach to the multiple-minima problem in protein folding. Proc. Natl. Acad. Sci. **84**(19), 6611–6615 (1987)
12. Mayuko, T.S., Daisuke, T., Chieko, C., Hirokazu, T., Hideaki, U.: Protein structure prediction in structure based drug design. Curr. Med. Chem. **11**(5), 551–558 (2004)
13. McGuffin, L.J., Bryson, K., Jones, D.T.: The PSIPRED protein structure prediction server. Bioinformatics **16**(4), 404–405 (2000)
14. Paluszewski, M., Hamelryck, T., Winter, P.: Reconstructing protein structure from solvent exposure using tabu search. In: Algorithms for Molecular Biology (ALMOB) (2006)
15. Paluszewski, M., Winter, P.: EBBA: efficient branch and bound algorithm for protein decoy generation. Technical report. Department of Computer Science, Univ. of Copenhagen, vol. 08(08) (2008)
16. Paluszewski, M., Winter, P.: Protein decoy generation using branch and bound with efficient bounding. In: Proc. of the 8th Int. Workshop, WABI 2008, LNBI 5251, pp. 382–393 (2008)

17. Pham, D., Koc, E., Ghanbarzadeh, A., Otri, S., Rahim, S., Zaidi, M., Phrueksanant, J., Lee, J., Sahran, S., Sholedolu, M., Ridley, M., Mahmuddin, M., Al-Jabbouli, H., Darwish, A.H., Soroka, A., Packianather, M., Castellani, M.: The Bees Algorithm—a novel tool for optimisation problems. In: Proceedings of IPROMS 2006 Conference, pp. 454–461 (2006)
18. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm. Technical report, MEC, Cardiff University, UK (2005)
19. Rohl, C.A., Strauss, C.E., Misura, K.M., Baker, D.: Protein structure prediction using Rosetta. Methods Enzymol. **383**, 66–93 (2004)
20. Sayle, R.: RasMol v2.5 a molecular visualisation program. Biomol. Struc. Glaxo Research and Development Greenford. Roger Sayle and Biomol. Struct. (1994)
21. Simons, K.T., Kooperberg, C., Huang, E., Baker, D.: Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J. Mol. Biol. **268**(1), 209–25 (1997)
22. Skolnick, J., Kolinski, A., Ortiz, A.R.: MONSSTER: a method for folding globular proteins with a small number of distance restraints. J. Mol. Biol. **265**, 217–241 (1997)
23. Song, J., Takemoto, K., Akutsu, T.: HSEpred: predict half-sphere exposure from protein sequences. Bioinformatics **24**, 1489–1497 (2008)
24. Vilhjalmsson, B., Hamelryck, T.: Predicting a new type of solvent exposure. In: ECCB, Computational Biology Madrid 05, P-C35, Poster (2005)
25. Yuan, Z.: Better prediction of protein contact number using a support vector regression analysis of amino acid sequence. BMC Bioinf. **6**(1), 248 (2005)
26. Zemla, A., Venclovas, C., Moult, J., Fidelis, K.: Processing and analysis of CASP3 protein structure predictions. Proteins **Suppl 3**, 22–29 (1999)
27. Zhang, Y.: I-TASSER server for protein 3D structure prediction. BMC Bioinf. **9**(1), 40 (2008)