Datalogisk Institut
KØBENHAVNS UNIVERSITET

DIKU

Protein Structure Prediction Using Bee Colony Optimization Metaheuristic

R. Fonseca, M. Paluszewski and P. Winter

Technical Report no. 08-12
ISSN: 0107-8283

Dept. of Computer Science
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Protein Structure Prediction Using Bee Colony Optimization Metaheuristic

R. Fonseca[1], M. Paluszewski, and P. Winter[1]

Dept. of Computer Science, Univ. of Copenhagen, Universitetsparken 1, 2100 Copenhagen Ø, Denmark
{ hite, palu, pawel }@diku.dk

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 is a new metaheuristic approach to optimization based on the foraging behaviour of bees. We have implemented the Bee Colony Optimization metaheuristic using hill-climbing as local search to generate good solutions to the protein structure prediction problem. With this method the choice of local search method can easily be changed, new solutions could be generated using evolutionary algorithms or the heuristic could be used to prioritize parallel runs of searches. The results show that Bee Colony Optimization generally finds better solutions than simulated annealing in the same amount of time.

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 has applications in medicine within areas like drug- and enzyme design [1].

The 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 when using 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 [2].

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 by [3] and was based on the mating of bee drones with a queen bee. The foraging approach was proposed simultaneously in [4] and [5] 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. [6] previously used the *Bees Algorithm* [4] 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 similarity measure, are generated. These decoy solutions 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 [7] or I-Tasser [8]. However, the BCO

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

metaheuristic has appealing properties such as local extremum avoidance and resource allocation for local searches, and we believe this makes it suitable for the PSP.

In section 2 the model of PSP and the energy function is defined. In section 3 our adaptation of BCO is described. Finally, experiments are described in section 4 and discussed in section 5.

2 Protein Structure Prediction

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

Proteins are chains of amino acids. There are 20 different kinds of amino acids, each represented by a letter. The sequence of amino acids is called the primary structure of the protein. Frequent occuring local structures of amino acids, such as helices and strands, are called secondary structure and the full description of the protein (i.e. 3D coordinates of all atoms) is called the tertiary structure. The protein representations described here are able to represent the tertiary structure of proteins.

All amino acids consist of identical 'backbones' (nitrogen and two carbon) and a side chain denoted R. One amino acid (glycine) only contains a single hydrogen atom in the side chain and therefore requires no parameters to represent R. Others have up to 18 atoms in the side chain and can require up to 5 *rotamer* angles (χ_{1-5}) to be fully represented.

Bonded to the backbone atoms are two hydrogen atoms and an oxygen atom. The chemical bonds within the backbone fixate the six atoms from (including) C^α in one amino acid to (including) C^α in the next on a planar rhombus (see Figure 1). The backbone structure of each amino acid can therefore be represented using two angles: $\varPhi$ and $\varPsi$. This is the representation used by Bahamish et al. [6].

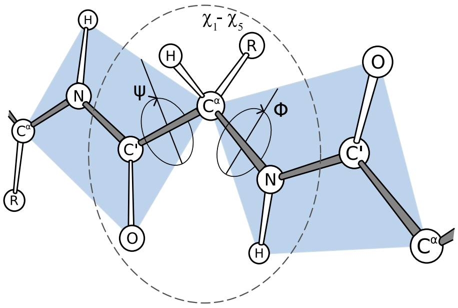

Fig. 1. The atoms and side chain of an amino acid (within the dotted line). The backbone is specified by the torsion angles $\varPhi$ and $\varPsi$, and the side chains by rotamer angles χ_1 to χ_5.

2.1 Segment representation

When trying to determine the overall 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 a protein is represented. This leads to the C^α-trace representation of proteins illustrated in Figure 2. Each amino acid can be represented by two angles, 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^α atoms in the tertiary structure (see Figure 3). Strands, additionally, are characterized by pairing up with strands different places in the protein. Coil is the class of all other shapes that are neither helices nor strands. C^α-atoms of a coil therefore have a large degree of freedom, compared to helices and strand, since there are few geometric constraints on the tertiary structure of a coil.

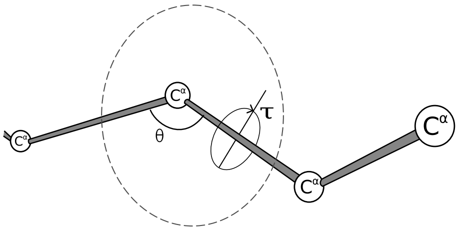

Fig. 2. C$_\alpha$ trace of backbone. Each amino acid is here specified by two angles θ and τ. The graphics are generated by Rasmol [9].

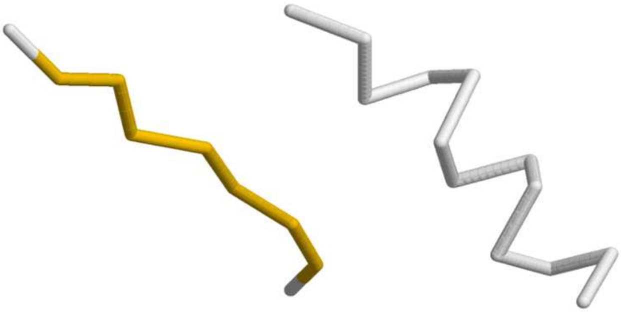

Fig. 3. Typical backbone structure for a strand (left) and a helix (right)

A sequence of residues of the same secondary structure class is here called a *segment*. Segments can be considered as rigid rods that define the overall path of C$^\alpha$-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 therefore an abstract representation of a sequence of residues and it does not explicitly contain the coordinates of internal C$^\alpha$-atoms. A segment structure is therefore defined to be the coordinates of all C$^\alpha$-atoms of a segment. The list of all segment structures is called the *complete structure*. Figure 4 is an illustration of a complete structure in the simplified 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 is reduced. Segments are therefore only allowed to have a discrete amount of predefined directions (d) between the first and last C$^\alpha$-atoms. Obviously, the chance of being able to represent a complete structure similar to the native structure of the protein increases the more when more directions are allowed. To further discretize the model, the number of possible segment structures allowed by a segment is limited to s. The method used to determine the structures of helix, strand and coil-segments is described in section 2.2.

Ad-hoc experiments show that $d = 73$ uniformly distributed directions acquired by combining the face centered cubic (FCC) lattice, the simple cubic (SC) lattice and the body centered cubic (BCC) lattice is suitable for representing realistic proteins. Experiments also show that allowing $s = 16$ structures seems suitable for BCO.

Given an amino acid sequence with m segments, d possible segment directions and s possible segment structures for each segment, the total number of complete structures, N, allowed by this model is limited by

$$N < d^m \cdot s^m$$

One might think that this should be $N = d^m \cdot s^m$, but because of rotational and mirror symmetry many complete structures can be disregarded. For instance the first segments direction and structure can be fixed, and in some cases the directions of the second segment also results in symmetrical structures that can be ignored.

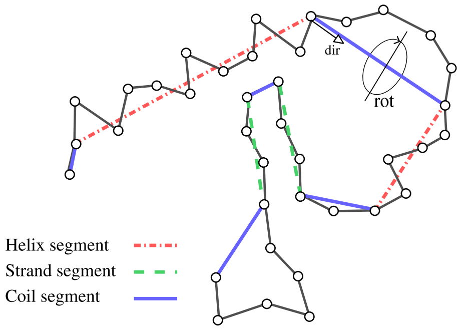

Fig. 4. Segment representation of proteins. Each segment can point in 73 directions and the amino acids can assume 16 distinct rotations around the segment-line

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 right-handed helix is the most commonly observed secondary structure in proteins. In helices, the most observed angle pair for an amino acid is $(\theta, \tau) = (91°, 49°)$. Given a helix segment, one segment structure having these angle properties are 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$^{\alpha}$-atoms.

Strand structures are constructed in the same way as helices, but with other angle values. For strands, the most observed angle pair is $(\theta, \tau) = (120°, 163°)$. The angle values were found after using P-SEA [10] to compute secondary structure of 3080 proteins from PDB Select (25) [11].

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 [12]. Given a coil segment, PDB Select (25) is queried with protein sequences and their known structures and find the $\sqrt{s}$ best fragment matches in terms of amino acid similarity. Each of these structures are rotated uniformly $\sqrt{s}$ times, as for helices and strands, such that a total of s structures are obtained. The fragment database does of course not contain the proteins used in the experiments.

2.3 Formal representation

A complete structure is defined by a discrete value of direction and structure for each segment. The complete structure of a protein with m segments is therefore specified by a list of directions and a list of structures:

$$\begin{aligned} &d_i && i = 1 \dots m, && d_i \in \{1, 2 \dots d\} \\ &s_i && i = 1 \dots m, && s_i \in \{1, 2 \dots s\} \end{aligned}$$

2.4 Energy

Determining an energy function for protein structures that is computationally fast and correlates well to the real native structure of proteins is still an open problem within bioinformatics. Some energy functions are based on quantum mechanical interactions between atoms of the protein, and, although the quality of the minimum energy structures is good, the computation of the energy usually takes a long time. Other energy functions – pseudo-energy-functions – are based on statistical analysis of large sets of proteins. These are usually very fast but the quality of the minimal energy structure varies greatly.

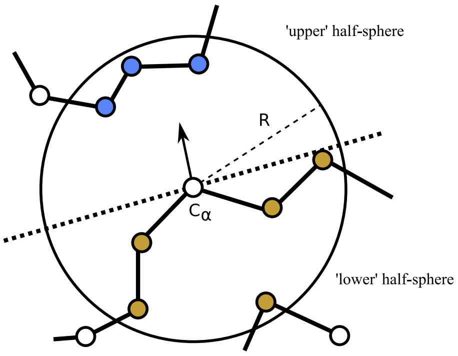

Fig. 5. Half-sphere exposure for an amino acid. The up/down pair is $(3, 5)$. The contact number is 8.

A promising pseudo-energy-function described in [13] is based on *Half-Sphere Exposure* (HSE) [14] and *Contact Numbers* (CN). This energy requires very little computation and represents many of the important properties of protein native structures.

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 Figure 5). The up vector relative to some amino acid A_i can be defined as

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

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. CN for an amino acid is the number of amino acids contained in the *entire* sphere. For each amino acid the HSE-pair and CN can be predicted from the primary structure alone using support vector regression [15, 16].

Let $\mathcal{P}$ denote the conformational space of a protein with n residues $A_1, A_2, ..., A_n$. Let $p \in \mathcal{P}$. The total energy $Q(p)$ is defined as the sum of the residue energy contributions $Q_p(A_i)$, i.e.,

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

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

where

- $\Delta CN(A_i)$ is the difference between the contact number of the i-th residue A_i in p and the desired (i.e., predicted by support vector regression) contact number of A_i .
- $\Delta HD(A_i)$ is the difference between the down half sphere exposure number of A_i in p and the desired down half sphere exposure number of A_i .
- $\Delta HU(A_i)$ is the similar difference for the up half sphere exposure.

The reason why CN instead of HSE is used for the first residue of a segment is that it was necessary for the Branch and Bound algorithm described in [13, 17]. In order to compare solutions found here with those in [13] the same energy function is preserved.

A radius of the contact sphere around 13Å is known to give a good prediction quality [18] 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 radius of the surrounding sphere – the *radius of gyration* (*Rg*)– is introduced. *Rg* can be predicted from the number of residues n of the protein [19]:

$$Rg = 2.2n^{0.38} \tag{1}$$

This prediction is often accurate for globular proteins. Infinite energy is therefore assigned to structures having radius of gyration more than 20% away from the predicted Rg. A structure is said to be clashing if the distance between two C$^{\alpha}$ atoms is less than 3.5Å. 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. 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 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 finding a neighborhood solution. The onlookers choose sites for neighborhood search based on the objective value of scouts and workers in previous iterations. This method is largely the *Bees Algorithm* proposed in [4]. In a non-changing solution space a solution does not deplete in the same way a real life flowerbed depletes of nectar. Exhaustion is therefore forced when a solution cannot be improved. This idea is somewhat similar to the idea of pruning parts of the search space as described in [20]. The process of exhausting a local search is proposed as part of the *Artificial Bee Colony* algorithm described in [5]. Our adaptation of the BCO metaheuristic is a synthesis of these approaches.

Algorithm 1: Bee-Colony-Optimization

```
input  : $S$, $W$, $O$, $Exhaust$, $OS$, $NS$, $SS$
output: The best solution
Initialize population with $S + W$ random solutions using $SS$
Evaluate cost of the population
while Stopping criterion is not met do
    Recruit $O$ onlooker-bees and assign each to a member of the population according to $OS$
    for Each onlooker assigned to some member $n$ of the population do
        Perform an iteration of the local search algorithm $NS$ on $n$
    end
    Evaluate cost of the population
    If a member of the population has not improved for $Exhaust$ iterations, save the
    solution and replace it with a random solution using $SS$
    Find $S$ random solutions using $SS$ and replace the $S$ members of the population that
    has the worst costs
end
return The best solution – either from the population or from the saved solutions
```

Here S, W and O is the amount of scout, worker and onlooker bees respectively. OS is the strategy for assigning onlookers, NS is the neighborhood strategy for performing a local search and SS is the method for generating a random solution.

3.1 Bee Colony Optimization applied to PSP

The above pseudocode can be used for any optimization problem where OS, NS and SS can be defined. So to utilize BCO for PSP these three methods have to be defined.

Scout search strategy (SS) 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 (OS) The onlookers choose a member n of the population based on the members energy function. If the member has a low energy then it is more likely to be chosen. This is implemented by letting each onlooker choose the member with highest estimated fitness:

$$fitness_n = \textsc{RandomNumberBetween}(0,1) \cdot \frac{1}{\textsc{Energy}(n)}$$

Onlookers Local Search (NS) Any local search could be utilized as neighborhood strategy so a simple hill-climbing strategy is chosen. Each iteration finds a random neighbor to the existing solution and replaces the existing solution if the energy is improved. The neighbor is generated by randomly changing two randomly chosen segments directions d_i, as well as four randomly chosen segments structure s_i.

4 Experiments and results

The tertiary structures of 8 proteins is predicted. 6 proteins have previously been used for benchmarks in the literature [21, 13, 22]. The remaining 2 are somewhat bigger and were chosen from the targets of CASP7. We have intentionally chosen a pair that proved to be hard to predict by CASP7 participants. Most succesful 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. The tertiary structures of the proteins are known and the quality of the results can therefore be evaluated using the Global Distance Test measure (GDT) [23]. $\mathrm{GDT}_c(p)$ 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 RMSD of the set is less than c. $\mathrm{GDT}(p)$ is defined as the average of $\mathrm{GDT}_1(p)$, $\mathrm{GDT}_2(p)$, $\mathrm{GDT}_4(p)$ and $\mathrm{GDT}_8(p)$.

The input to BCO is a secondary structure assignment, HSE-vector 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 [24] and HSE-vectors using LAKI [18] and HSEpred [15]. For better comparison of energy levels, the HSE predictions from [13], which were done using LAKI [18], were used. For the CASP proteins the newer and more accurate HSE prediction server HSEpred [15] were used. Note that PSIPRED, LAKI and HSEpred are neural networks trained on a selection of proteins from PDB. The 8 benchmark proteins used here also exist in PDB, so there is a slight chance that the training sets for PSIPRED, LAKI and HSEpred contain some of these proteins. However, the prediction quality of the 8 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. The radius of gyration is predicted using Equation 1.

For comparison and evaluation of the model and prediction quality, all experiments are also done using the exact secondary structures and exact HSE-vectors obtained from the native structures of the proteins. These structures cannot be considered solved *de novo*. All computations were performed on a 3.4GHz Intel Xeon with 2GB RAM.

By ad-hoc experiments an appropriate configuration for BCO was determined. $S = 10$ scouts, $W = 10$ workers and $O = 100$ onlookers were used, $Exhaust$ was set to 5 and the algorithm was set to stop when it had run for 48 hours. Since the purpose of the BCO algorithm is to find many good decoys the best 1000 unique solutions are registered.

To evaluate BCO as an optimization metaheuristic it is compared to simulated annealing (SA) by running 10 parallel instances of SA in 48 hours in total on every protein. The SA algorithm also stores 1000 unique registered decoy solutions with minimal energy. A solution is registered if it is encountered at some point in one of the 10 searches. The results from EBBA [13] are also presented here for comparison. Even though the representation in [13] is the same as here, some parameters diverge, namely the amount of segment directions d (12 in [13], 73 for BCO) and structures s (2 to 8 in [13], 16 for BCO). Also the tolerated divergence from the predicted radius of gyration differs (5% in [13], 20% here).

Table 1 summarizes the results of the runs from BCO, SA, EBBA and CASP7. p^* is the protein structure encountered during a search for which the energy function $Q(p)$ is lowest. For BCO, SA and EBBA this energy function is identical. $p^\dagger$ is the protein structure – among the 1000 saved decoys – for which $\mathrm{GDT}(p)$ is highest.

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%	-	-	-

Table 1. Results from Bee Colony Optimization (BCO), Simulated Annealing (SA), Efficient Branch and Bound Algorithm (EBBA) and CASP7. At CASP7 the proteins 2HG6 and 2J6A had target numbers T0314 and T0319 respectively. Large values of GDT are preferrable 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.

5 Discussion and conclusion

The results of BCO, 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 energy function does not identify the best structure since GDT(p^*) is relatively low for BCO and SA. Assuming, however, that a more advanced energy function can identify $p^\dagger$, this would rank the structures obtained by BCO as $17 - th$ of 132 for 2J6A and $30 - th$ out of 132 for 2HG6 at CASP7.

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 this type of problems. The average values of $Q(p^*)$ for the 6 smaller proteins are illustrated in Table 2. For these proteins BCO finds values of $Q(p^*)$ that, on average, is 5% better than those found by SA. It is worth noting that SA usually is the algorithm of choice when choosing a metaheuristic for PSP.

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

Table 2. Comparison of optimal encountered energy values for BCO, SA and EBBA when run on 1FC2, 1ENH, 2GB1, 2CRO, 1CTF and 4ICB . Note that some parameters diverge in EBBA's representation of the protein and EBBA is the only algorithm that guarantees a globally optimal p^*.

EBBA is an exact algorithm that guarantees to find the structure with minimal energy, yet $Q(p)$ is higher than the energy BCO finds because more segment directions and rotations are allowed in BCO and SA.

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 wich 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 contains enough information to identify the native structure of the protein. There are two possible 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.

References

1. T. S. Mayuko, T. Daisuke, C. Chieko, T. Hirokazu, and U. Hideaki. Protein structure prediction in structure based drug design. *Current Medicinal Chemistry*, 11(5):551–558, 2004.
2. Z. Li and H. A. Scheraga. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci.*, 84(19):6611–6615, Oct. 1987.
3. H. A. Abbass. MBO: Marriage in honey bees optimization - A haplometrosis polygynous swarming approach. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 207–214, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 2001. IEEE Press.
4. D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. The bees algorithm. Technical report, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
5. D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes Univ., Engineering Faculty, Computer Engineering Department, Nov. 2005.
6. H. A. A. Bahamish, R. Abdullah, and R. A. Salam. Protein conformational search using bees algorithm. In *Asia International Conference on Modelling and Simulation*, pages 911–916. IEEE Computer Society, 2008.
7. C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker. Protein structure prediction using Rosetta. *Methods Enzymol*, 383:66–93, 2004.
8. Y. Zhang. I-tasser server for protein 3d structure prediction. *BMC Bioinformatics*, 9:40+, Jan. 2008.
9. R. Sayle. RasMol v2.5 A molecular visualisation program, Biomolecular Structure Glaxo Research and Development Greenford, 1994. Roger Sayle and Biomolecular Structure.

10. G. Labesse, N. Colloc'h, J. Pothier, and J.-P. Mornon. P-SEA: A new efficient assignment of secondary structure from C alpha trace of proteins. *Bioinformatics*, 13:291–295, 1997.
11. U. Hobohm and C. Sander. Enlarged representative set of protein structures. *Protein Sci*, 3(3):522–524, 1994.
12. C. Chothia and A. M. Lesk. The relation between the divergence of sequence and structure in proteins. *The EMBO Journal*, 5:823–826, 1986.
13. M. Paluszewski and P. Winter. Protein decoy generation using branch and bound with efficient bounding. *Proc. of the 8th Int. Workshop, WABI 2008, LNBI 5251*, pages 382–393, 2008.
14. T. Hamelryck. An amino acid has two sides: A new 2d measure provides a different view of solvent exposure. *J. Proteins: Structure, Function, and Bioinformatics*, 59(1):38–48, 2005.
15. J. Song, K. Takemoto, and T. Akutsu. HSEpred: Predict half-sphere exposure from protein sequences. *Bioinformatics*, 24:1489–1497, 2008.
16. Z. Yuan. Better prediction of protein contact number using a support vector regression analysis of amino acid sequence. *BMC Bioinformatics*, 6(1):248, 2005.
17. M. Paluszewski and P. Winter. EBBA: Efficient branch and bound algorithm for protein decoy generation. *Technical report. Department of Computer Science, Univ. of Copenhagen*, 08(08), 2008.
18. B. Vilhjalmsson and T. Hamelryck. Predicting a new type of solvent exposure. ECCB, Computational Biology Madrid 05, P-C35, Poster, 2005.
19. J. Skolnick, A. Kolinski, and A. R. Ortiz. MONSSTER: A method for folding globular proteins with a small number of distance restraints. *J. Mol. Biol.*, 265:217–241, 1997.
20. M. Paluszewski, T. Hamelryck, and P. Winter. Reconstructing protein structure from solvent exposure using tabu search. *Algorithms For Molecular Biology (ALMOB)*, 2006.
21. T. Hamelryck, J. T. Kent, and A. Krogh. Sampling realistic protein conformations using local structural bias. *PLOS Computational Biology*, 2, 2006.
22. K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. 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.
23. A. Zemla, C. Venclovas, J. Moult, and K. Fidelis. Processing and analysis of CASP3 protein structure predictions. *Proteins*, Suppl 3:22–29, 1999.
24. L. J. McGuffin, K. Bryson, and D. T. Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.