Bonding as a Swarm: Applying Bee Nest-Site Selection Behaviour to Protein Docking

Konrad Diwold
Faculty of Mathematics and Computer Science
Universität Leipzig
Leipzig, Germany
kdiwold@informatik.uni-leipzig.de

Daniel Himmelbach
Faculty of Mathematics and Computer Science
Universität Leipzig
Leipzig, Germany
baarlun@googlemail.com

René Meier
Faculty of Biosciences, Pharmacy and Psychology
Universität Leipzig
Leipzig, Germany
rene.meier@uni-leipzig.de

Carsten Baldauf
Fritz-Haber-Institut der Max-Planck-Gesellschaft
Berlin, Germany
baldauf@fhi-berlin.mpg.de

Martin Middendorf
Faculty of Mathematics and Computer Science
Universität Leipzig
Leipzig, Germany
middendorf@informatik.uni-leipzig.de

ABSTRACT

The identification of protein binding sites and the prediction of protein-ligand complexes play a key role in the pharmaceutical drug design process and many domains of life sciences. Computational approaches for protein-ligand docking (or molecular docking) have received increased attention over the last years as they allow inexpensive and fast prediction of protein-ligand complexes. Here we introduce the principle of Bee Nest-Site Selection Optimisation (BNSO), which solves optimisation problems using a novel scheme inspired by the nest-site selection behaviour found in honeybees. Moreover, the first BNSO algorithm – called Bee-Nest – is proposed and is applied to the protein-docking problem. The performance of Bee-Nest is tested on 173 docking instances from the PDBbind core set and compared to the performance of three reference algorithms. The results show that Bee-Nest could find ligand poses with very good energy levels. Interestingly, the reference Particle Swarm Optimization (PSO) produces results that are qualitatively closer to wet-lab experimentally derived complexes but of worse energy than Bee-Nest. Our results clearly highlight the superior performance of Bee-Nest in semi-local (regional) optimization for the molecular docking problem, and suggests its usefulness in a hybrid strategy.

Keywords

Swarm Intelligence, Bee-Inspired optimization, Protein Docking

GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0557-0/11/07 ...$10.00.

1. INTRODUCTION

By recognizing small molecules, proteins act as the receptors of ligands. These interactions are formed if the three dimensional (3D) structure of the ligand fits into the binding pocket of the protein, like a key into a lock (see Figure 1 for an exemplary illustration). Knowledge about such interactions is crucial for the understanding of physiological processes and is a fundamental basis for the development of pharmaceutical substances.

The 3D structural information has been experimentally resolved for only a limited number of protein-ligand pairs, while no such data is available for the vast majority of cases. As resolving structural ligand-protein information experimentally is quite costly, computational approaches have become more and more established in the prediction of such complexes [4]. Computational approaches allow the fast and inexpensive screening of large libraries of potential ligands against a variety of protein targets and thus serve as a means of sampling potential ligand candidates, with the best results then being further investigated in wet-lab experiments. Such rapid *in silico*-screening methods are of growing importance in the industrial drug design process.

From a computer science perspective, molecular docking boils down to an optimization problem, namely finding the protein-ligand pose with minimal binding energy. Given a scoring function that estimates the binding energy of a protein-ligand complex, the docking problem results in the search for the global minimum in a multi-dimensional energy landscape.

Several population-based metaheuristics such as genetic algorithms [7], ant colony optimization [10] and particle swarm optimization [11, 5, 13] (PSO) have been proposed to provide solutions to the problem of protein-docking. In this paper a novel population-based metaheuristic for protein docking is outlined: the Bee Nest-Site optimization algorithm (BNSO), which is based on the nest-site selection process in honeybees.

Honeybees and the mechanisms underlying their collective

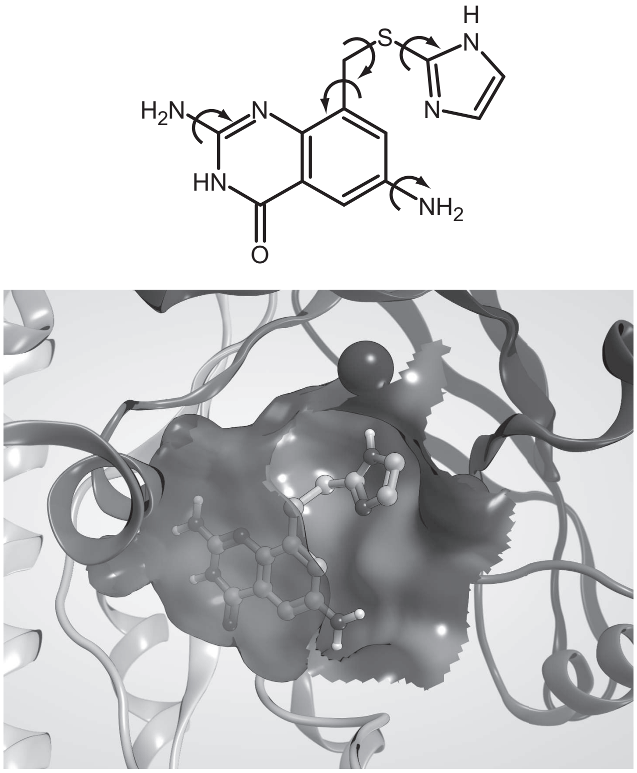


Figure 1: A small molecular ligand visualised as a chemical formula with internal degrees of freedom highlighted by arrows (top) and the ligand bound into the pocket of a tRNA-Guanine Transglycosylase (bottom) as experimentally resolved by X-ray crystallography (Protein Database entry 1K4G)

behaviour have recently served as a blueprint for several algorithms in the field of swarm intelligence. Bee-inspired algorithms have been proposed as solutions for problems as diverse as network routing [3], optimization [9], and robotics [16].

In the context of optimization, existing bee-inspired algorithms are usually based on one of the following two behavioural concepts: the mating behaviour or the foraging behaviour (see [9] for an in-depth review on bee-inspired optimization approaches). A recent study [1] outlined the optimization potential of another behaviour found in honeybees: nest-site selection. Nest-site selection constitutes a decentralized decision-making process in honeybees, where a swarm of homeless bees has to identify and choose a new nest-site among several potential sites. On the basis of a biological model, Diwold et al. [1] proposed a general scheme, called Bee Nest-Site Selection Scheme, that can be used for developing algorithms based on the nest-site selection behaviour of honey bees. As a proof of concept it was shown that the Bee Nest-Site Selection Scheme can select good nest-sites in dynamic and noisy environments, and that its iterative application can lead to an optimization process converging towards an optimum. Applying the Bee Nest-Site Selection Scheme for optimization is called Bee Nest-Site optimization (BNSO) here and the corresponding optimization algorithms are called Bee Nest-Site optimization algorithms (BNSO algorithms).

In this paper we develop the first Bee Nest-Site optimization algorithm called Bee-Nest for solving a complex optimization problem. Bee-Nest is applied to the molecular docking problem. The algorithm is realized in ParaDocks, a molecular docking framework specially developed for population based heuristics [13] [1]. To benchmark the performance of the Bee-Nest for the docking problem, instances from the PDBbind database *core-set* [20] were used for testing. The obtained optimization and sampling performance was compared to a PSO algorithm that was previously proposed for docking [13], as well as randomly selected, solutions and solution derived using local optimization.

This paper is structured as follows. In Section 2 the biological background of nest-site selection in honeybees is described. Section 3 introduces the BNSO principle and the Bee-Nest algorithm. The experimental setup, the docking framework, and the data-set used for benchmarking are described in Section 4. Results are presented in Section 5 and concluding remarks are given in Section 6.

2. BEE NEST-SITE SELECTION

Nest-site selection is part of the reproductive cycle of honeybee swarms. When a bee colony reaches a certain size, its reproductive mechanisms are triggered by the production of new queens. While one of the new queens will inherit the established hive and two thirds of its worker population, the old queen and the remaining workers will leave in search of a new hive [18].

The old queen and her followers are called a reproductive swarm. They will settle in a temporary location shortly after leaving the hive. The majority of workers will stay with the queen shielding her from external threat by forming a tight cluster around her. A small fraction of the swarm (approximately 5%) will start the nest-site selection process. Scouts leave the swarm to search for suitable new nest-sites such as cavities in trees or buildings. Upon finding a potential nest-site, a scout will assess several aspects of the site such as its volume, height, and the size of the entrance in order to evaluate its quality [18].

Sites of sufficient quality will be advertised by the scouts upon their return to the swarm. A scout that has found a potential nest-site will advertise it with a waggle dance, which encodes the direction and distance to the site. The duration of the waggle dance encodes a site's quality. Bees that follow a waggle dance learn about the nest-site's location and will visit it after the dance for an independent evaluation.

When its dance is over, a scout will return to the nest-site it promoted and reassess it, then it returns to the swarm and readvertises it again. The number of dances a scout performs for the same nest-site over consecutive visits decreases at a constant rate, a phenomenon known as dance attrition [19]. While this process of quality-independent dance attrition prevents the swarm from becoming deadlocked in their decision [15], it will still ensure that better nest-sites are longer and more often danced for, which leads to more individuals being recruited to high-quality sites than low-quality sites over time.

During nest-site assessment, a scout also estimates the number of other scouts present at that given site. If the number of scouts exceeds a certain threshold (called quorum

[1] ParaDocks can be downloaded from http://www.paradocks.org

in this context) a decision has been made. Scouts present at the site will fly back to the swarm and initiate the swarm's lift-off.

The airborne swarm will then move towards the chosen site. The mechanism underlying the swarm's flight is still under debate. The best established hypothesis is that informed scouts guide the swarm towards a new location by flying rapidly through the swarm in the direction of the nest site [17]. Finally, after reaching the new nest-site the bees move in and establish a new hive.

3. ALGORITHM

Bee Nest-Site optimization (BNSO) is an application of Diwold et al.'s Bee nest-Site Selection Scheme [2] (BNSSS), which itself constitutes an optimization-oriented extension of the individual-based nest-site selection model for honey bees developed by Janson et al. [6].

A Bee-Nest optimization starts with a colony of virtual bees being placed at a random position in search space. Here the search space represents an environment and each position in the search space corresponds to a potential nest-site (solution). The quality of a nest-site is given by the value of the function to be optimized at the corresponding position.

Using the principles of nest-site selection the colony tries to find a nest-site of better quality than its current location. A colony contains two types of bees: scouts and followers. The selection process begins with scouts trying to find potential nest-sites in the surroundings of the swarm's current location. If the scouts are able to find a location that is of acceptable quality, they report it to the swarm. Followers choose a scout to follow based on the quality of the nest-site it has found (i.e., scouts that found better nest-sites will attract more followers). The follower then flies to the location the scout found and searches the surrounding to eventually find a better location. If the colony is able to come up with a location that is of better quality than its current location it will relocate itself to the new location and restart the nest-site selection process. Otherwise, the colony repeats the selection process at its current location.

More formally: Given a dim dimensional function F that is to be minimized and a swarm of n virtual bees consisting of n_{scout} scouts and $n_{follower}$ followers (i.e., $n = n_{scout} + n_{follower}$). The swarm is initially placed on a randomly chosen location $p_{swarm} = (x_1, \ldots, x_{dim})$ in the search space. Each scout s chooses uniformly at random a location p_s with the restriction that it has at most distance $d_{scout} \times f_r$ to the swarm's current location (i.e., $|p_{swarm} - p_s| \leq d_{scout} \times f_r$). Here d_{scout} is a parameter and f_r $(0 \leq f_r \leq 1)$ is a factor that decreases over time in order to achieve an increasingly local search of the algorithm in the course of the optimization. One possibility of defining f_r is to predefine a maximum number of iterations $MAXITER$ per optimization run and adapting f_r accordingly by

$$f_r = 1 - \frac{iteration}{MAXITER} \tag{1}$$

where *iteration* is the number of the current iteration. It is checked if the quality of the chosen location is sufficient such that $F(p_s) \leq F(p_{swarm}) \times f_q$, where parameter f_q $(0 \leq f_q \leq 1)$ is a quality factor. In that case the scout has found a potential nest-site at location p_s, which can then be chosen by the followers. The probability to be chosen by a follower depends on its relative fitness defined by $fit_s = \max\{0, (F(p_{swarm}) \times f_q) - F(p_s)\}$.

Algorithm 1 Bee-Nest

```
place swarm on random location $p$, i.e., $p_{swarm} = p$
repeats = 0;
**while** stop criterion not satisfied **do**
    reduce $f_{range}$ according to Eq. 1
    **for all** scouts **do**
        Choose new location $p_s$ with a max distance of $d_{scout} \times f_{range}$ to the nest
        $fit_s = \max\{0, (F(p_{swarm}) \times f_q) - F(p_s)\}$
    **end for**
    **for all** followers **do**
        Choose a scout $s$ according to Eq. 2
        Choose new location $p_{follower}$ with a max distance of $d_{follower}$ to chosen scouts position $p_s$
        Sample search space between $p_s$ and $p_{follower}$ in $m$ flight steps
    **end for**
    **if** better location $p$ was found **then**
        Relocate swarm to $p$, i.e., $p_{swarm} = p$
    **else**
        **if** $repeats \geq MAXREPEATS$ **then**
            Place swarm on new random location $p$, i.e., $p_{swarm} = p$
            repeats=0;
        **else**
            repeats++;
        **end if**
    **end if**
**end while**
```

After each scout has updated its location, each follower f chooses one scout using a standard roulette wheel selection so that the probability P_s of choosing scout s is

$$P_s = \frac{fit_s}{\sum_{k=1}^{n_{scout}} fit_k}. \tag{2}$$

Each follower is then placed to the location of the scout it has chosen. Then the follower chooses uniformly at random a location p_f in the vicinity of the scout's location p_s such that it is not further distant than a $d_{follower}$ (i.e., $|p_s - p_f| \leq d_{follower}$) where $d_{follower}$ is a parameter. Then the follower samples the search space between p_s and p_f in a directed flight consisting of m equal length flight steps where at the end of each step function F is evaluated.

During the whole process the system maintains the best solution p_{best} found so far. If the swarm is able to find a better location than its current location (i.e., $F(p_{best}) > F(p_{swarm})$) it migrates to the new location. Otherwise it restarts the nest-site selection process from its current location. If a swarm is not able to improve its location in $MAXREPEATS$ nest-site selection attempts it is moved to a random location in the search space and the nest-site selection process restarts. The algorithm terminates when a given stopping criterion is satisfied.

For a better understanding a pseudocode of Bee-Nest is presented in Algorithm 1. A visualization of a search run over 4 nest-site relocations, containing the search trajectories of the scouts and the followers, on a two dimensional Sphere function is shown in Figure 2.

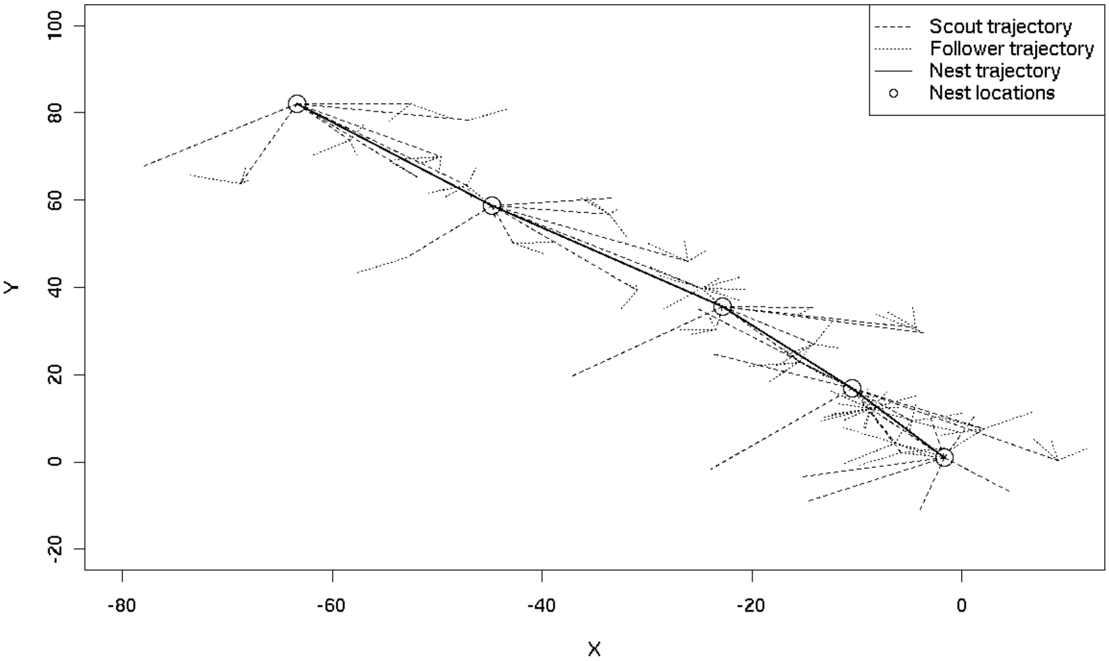


Figure 2: Visualization of a BNSO optimization run over 4 nest relocations (in this case on a two dimensional Sphere benchmark function)

4. EXPERIMENTAL SETUP

Modeling of molecules and molecular complexes in chemistry and biochemistry always features a variety of approximations, some of which affect the number of degrees of freedom (DOF), i.e., the dimension of the search space. In the approximation that used in this paper, a ligand-receptor pose is described by a vector containing three Cartesian coordinates for the ligand's position, its orientation is described by the three DOF of a quaternion, and the N internal DOF describe the ligand's conformation. In the test instances the internal flexibility ranges up to $N = 35$ internal DOF. Thus, the search space with $7+N$ dimensions, can be up to 42-dimensional. The conformation of the receptor is regarded to be rigid (this is an accepted approximation in the field). 173 instances from the PDBbind *core set* [20] database were used to test and compare the performance of the algorithms.

The statistically derived potential PMF04 was used to describe the binding energy landscape between ligand and receptor as pair-wise potentials of ligand and receptor atoms:

$$W_{ij}(d_{ij}) = -\ln \frac{g_{ij}(d_{ij})}{g_{ref}}, \qquad (3)$$

with $g_{ij}(d_{ij})$ the density of the atom pair ij in distance d_{ij} and g_{ref} the average density of atom pair ij. PMF04 is derived from 6611 protein ligand complexes and describes the interactions of 17 protein atom types with 34 ligand atom types. For a detailed description we point to the original publication by Muegge et al. [14]. The adaptations necessary to use PMF04 for molecular docking are described in [13].

The following three optimization algorithms were employed as a reference:

> PSO: The PSO was used with the settings suggested in [13] with 30 particles evaluated in 300,000 generations.
>
> RNDM: Nine million random poses were generated based on the Mersenne twister algorithm published by Matsumoto et al. [12], the best result was kept.
>
> RHC: 9,000 randomly chosen poses were locally optimized by 1,000 hill climbing steps. Lower energy poses are accepted, higher energy poses are discarded.

For the molecular docking problem the BNSO algorithm Bee-Nest was slightly extended with a local search as follows. When a better nest-site p_{nn} was found, by a scout or follower, a simple random walk, outlined below in Algorithm 2, was applied to the location for $MaxLO \geq 1$ times for the purpose of local optimization. This random walk generates for $MaxLO$ times a uniformly at random chosen location in the vicinity of the current best location p_{nn}. The maximum distance of the randomly generated location p_r to the current best location p_{nn} is restricted to $|p_r - p_{nn}| < f_l * d_{scout}$ where f_l is a parameter. Parameter f_l decreases over the steps of the local search towards 0 (details see Algorithm 2), this leads to the convergence of the new locations p_r to p_{nn}. The random walk is also applied to the final location returned by the BNSO for $PostLO$ times.

For the experimental runs of the BNSO algorithm the following parameter settings used: $n = 30$, $n_{scout} = 10$,

Algorithm 2 Random Walk

```
for $k \in 0 \dots MaxLO$ do
    $f_l = (1 - k/MaxLO)/16$
    Generate new random solution $p_r$ with $|p_r - p_{nn}| < f_l * d_{scout}$
    if $(F(p_r) < F(p_{nn})$ then
        $p_{nn} = p_r$
    end if
end for
```

$n_{follower} = 20$, $f_q = 0.95$, $MAXREPEATS = 20$, $MaxLO = 4$, $PostLO = 4096$. Since in the context of molecular docking, the dimensions of the search space correspond to different aspects of the problem (position, orientation, rotations of single axes in the molecules (internal DOF)) different values of d_{scout} ($d_{follower}$) are used for the different types dimensions in order to determine the range within to search for new locations around the current nest location (respectively, around the location of the scout):

$$d_{scout} = \begin{cases} 0.003616 \times SpaceRange, & \text{for position} \\ 0.001084 \times 2\pi, & \text{for orientation} \\ 0.027854 \times 2\pi, & \text{for internal DOFs} \end{cases}$$

$$d_{follower} = \begin{cases} 0.025366 \times SpaceRange, & \text{for position,} \\ 0.039257 \times 2\pi, & \text{for orientation} \\ 0.012289 \times 2\pi, & \text{for internal DOFs} \end{cases}$$

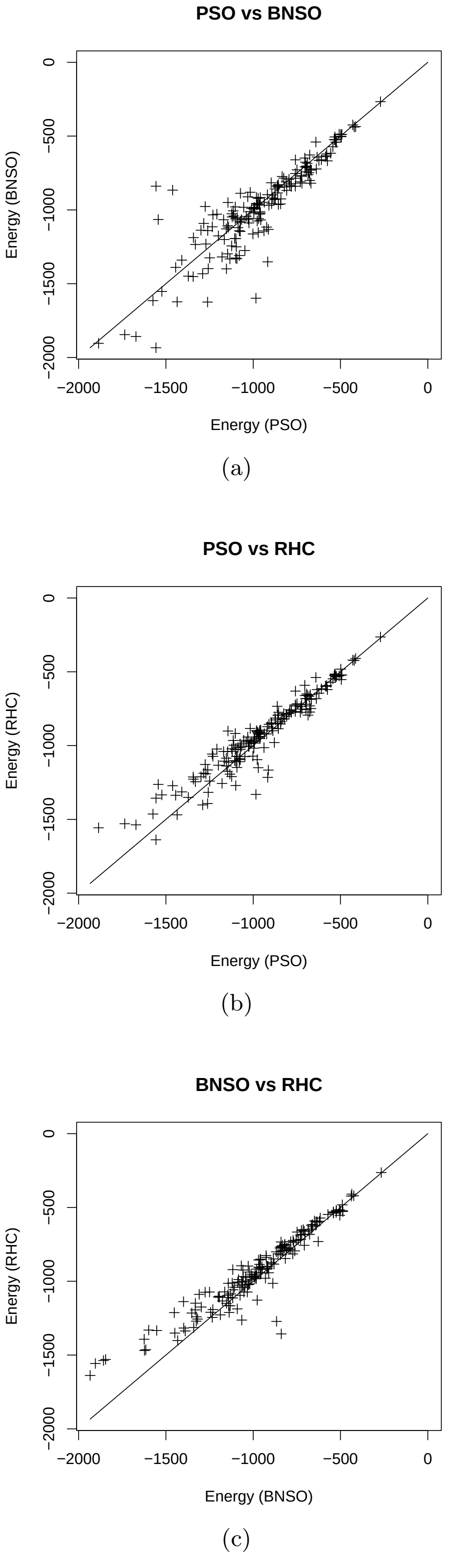
For each of the four algorithms every test for each of the 173 test instances $9,000,000$ energy evaluations have been done. Each test case was repeated 50 times. The test results allow to draw conclusions about the quality of the solutions and the robustness of the algorithms with regards to the molecular docking problem. The binding poses and the estimated energies resulting from each test run were recorded for each algorithm and are discussed in the next section.

5. RESULTS

Tables 1,2 and 3 show a comparison of minimum (respectively, median, mean) energy values achieved by the four algorithms on average over all test-instances. As can be seen, the BNSO algorithm Bee-Nest performs very well. It is able to achieve better energy values than PSO as well as RHC and RNDM on the majority of the test instances for all three criteria. The random hill climbing method (RHC) shows

Table 1: Minimum energy value comparison for all 173 test instances. Each cell denotes the number of test instances for which the minimum energy value obtained in 50 test runs of the algorithm stated in the row was better (i.e., lower) than the minimum energy of the algorithm stated in the column.

Alg vs. Alg	PSO	BNSO	RNDM	RHC
PSO	-	32	172	134
BNSO	141	-	173	168
RNDM	1	0	-	0
RHC	39	5	173	-



(a)

(b)

(c)

Figure 3: Scatter plots comparing the median performance (energy) of two algorithms for all test instances. Each data point indicates the contrasted algorithms' performances on a specific test instance. Points lying on the diagonal reflect comparable performance by each algorithm. As lower energy reflects better performance, points above the diagonal indicate better performance by the algorithm indicated on the x-axis, and vice versa.

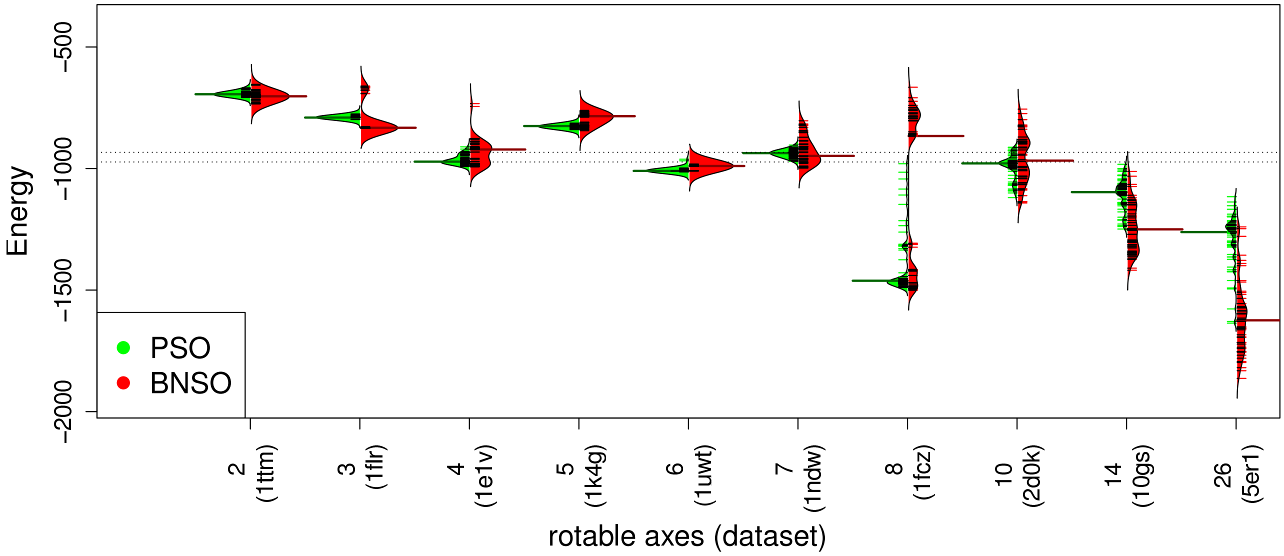


Figure 4: Bean plot of representative test instances from the test set. Each bean depicts the performance (energy) distribution of the PSO (green) and BNSO (red) arising from 50 repeats of each instance (x-axis). Instances are ordered by number of rotatable axes. Small coloured lines depict individual data points; dark lines show distribution mean.

Table 2: Median energy value comparison (analogously as in Table 1.

Alg vs. Alg	PSO	BNSO	RNDM	RHC
PSO	-	74	173	116
BNSO	99	-	171	142
RNDM	1	0	-	0
RHC	39	5	173	-

Table 3: Mean energy value comparison (analogously as in Table 1.

Alg vs. Alg	PSO	BNSO	RNDM	RHC
PSO	-	77	173	113
BNSO	96	-	173	133
RNDM	0	0	-	0
RHC	60	40	173	-

a decent performance, which is slightly worse than PSO and BNSO. The randomly generated solutions of RNDM are outperformed in each aspect by the other algorithms. Table 1 suggests that BNSO is especially capable of finding very low energy levels. In comparison with PSO it found the protein conformations with the lowest energy levels in 141 of the 173 test instances. Figure 3 depicts scatter plots of the median energy levels found by the BNSO, PSO, and RHC in all test instances. Scatter plots for the RNDM are omitted as its performance was very poor in general (see Tables 1-3). In Figure 3 the x-value corresponds to the median energy value of one algorithm for a given protein docking instance and the respective y-value corresponds to the energy value of a reference algorithm for the same docking instance. Values on or close to the diagonal denote test-cases where the algorithms showed a similar performance with reference to the energy levels. Values above the diagonal correspond to instances where the algorithm on the x-axis achieved better energy values and values below denote instances where the algorithm on the y-axis produced better energy values.

As can be seen in Figure 3(a) PSO and BNSO perform on par in instances with high energy levels (which usually corresponds to proteins with a small number of rotatable axes). In comparison to PSO the performance of BNSO improves for instances with a higher number of rotatable axes in the ligand. This can also be observed when comparing the BNSO with the RHC.

Figure 5 shows beanplots (see [8] for more details) depicting the estimated energy level distributions of the 50 solutions found by BNSO and PSO for a representative subset of docking instances from the test set. As can be seen the spread and thus the solution diversity increases with the increase of the internal flexibility of the ligand (number of rotatable axes). This is not surprising as this increases the dimensionality of the search space and thus leads to a more complex fitness landscape. In cases of an increased number of rotatable axes, the distribution of the PSO's energy levels is quite narrow in comparison to the BNSO's energy level distribution. This suggests that PSO generates protein ligand poses that are similar. Compared to that BNSO is more likely to produce a variety of poses during the 50 test runs, especially for proteins with many rotatable axes. Furthermore, the behaviour illustrated in the plots (Figure 5) also suggests that BNSO has some problems of escaping local optima. This is best seen for the docking instance $1fcz$, where the BNSO converges to either one of two possible minima, one with a suboptimal energy level around -800 and one with an optimal energy level around -1300. In contrast, PSO converges towards a configuration at the low energy level in most of the cases. However, such a spread can also be beneficial; for example in the docking instances $10gs$ and $5er1$, where BNSO is able to reach lower energy levels whereas the PSO is apparently stuck in suboptimal configurations.

5.1 Root mean square deviation

As pointed out above, the energy levels of the protein

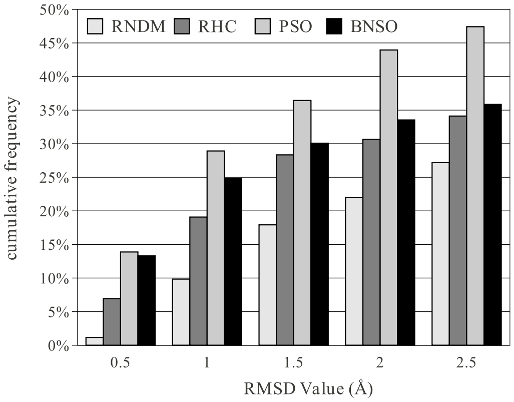


Figure 5: Cumulative histogram of the RMSD values from the X-ray crystal structures of the best solutions.

configurations found by BNSO in comparison to the reference algorithms are promising as they are in general of lower energy. In order to judge the biological significance of the calculated poses, the root mean square deviation (RMSD) was calculated for the best results found for each instance by each algorithm. RMSD is often used to measures the average distance between the different conformations of molecules. Here we compute the RMSD of the poses generated by our docking experiments with respect to wet-lab experimentally derived 3D structures resolved by X-ray crystallography. Thus, the RMSD value is a good estimate for the biological plausibility of the calculated conformation. RMSD gives the deviation of the generated protein-ligand pose from experimentally generated reference in $\mathring{A}$ngström ($0.1nm$). As proteins are not rigid bodies in space, RMSD values of up to $2.5\mathring{A}$ can be considered as a reasonable fit.

Figure 5 depicts a cumulative histogram of the RMSD values of the solutions generated by the four algorithms RNDM, RHC, PSO, and BNSO. The solutions produced by the PSO show the best fit regarding the real position and conformation of the ligand in the receptor. Whereas, BNSO and PSO produce roughly the same amount of conformations that are a very close fit (i.e., 13% and 14% of the poses produced by the BNSO and PSO, respectively, have an RMSD $\leq 0.5\mathring{A}$), this does not hold for higher RMSD values. Only 36% of the solutions found by BNSO have a RMSD value $\leq 2.5\mathring{A}$, whereas this is the case for 47% of the conformations produced by PSO. This result is unexpected, as it was shown in the last section that the energy levels of the conformation produced by BNSO are in general lower than those of PSO conformations.

There are two potential explanations for this observation: As outlined earlier, receptor-ligand conformations are evaluated using approximate energy functions to estimate their energy. Thus, part of the problem can come from the accuracy of the scoring function. It could, for example, be the case that the low-energy conformations found by the BNSO are not plausible in comparison to the real conformation. However, this can only explain a part of this odd behaviour, as this argument also applies to the solutions generated by the PSO. Another explanation for this phenomenon is that even though BNSO gets stuck in local optima sometimes it is still able to adapt the conformation of the ligand in such a way that it leads to low energy values. This would highlight the ability of BNSO to generate low-energy solutions, but also shows its limited ability to overcome larger energy barriers during the optimization process, as BNSO has a single position (i.e., receptor-ligand pose) as a starting point which is then continuously improved. In contrast, PSO performs a more thorough global search, as PSO starts off with its particles distributed in the whole search space. Molecular docking fitness landscapes are by no means a steady environment. Usually, only a very limited number of conformations yield low energy levels and seemingly small variations in the conformations can lead to a drastic quality change. It could thus be that while BNSO outperforms PSO in terms of fine-tuning the conformation of the protein-complex, it is not able to creep over the fitness barriers which are imposed by the fitness landscape as good as PSO. Both explanations will be further investigated in future work, for example by using different scoring functions. If the latter explanation turns out to be true, a hybrid approach in which SO is used to sample the search space and BNSO acts more as a fine-tuning mechanism, might yield an algorithm of truly improved performance.

6. CONCLUSIONS

This article introduced Bee Nest-Site optimization (BNSO) as a novel bio-inspired optimization principle based on the nest-site selection behaviour of honeybees. As shown in a previous study [2] the mechanism underlying nest-site selection can be useful in the context of optimization. Here an abstraction of the biological mechanism was outlined that can be applied in the domain of function optimization. The corresponding BNSO algorithm called Bee-Nest was tested in the domain of protein docking and its performance was compared to three reference algorithms (PSO, RHC, RNDM) that have been previously used in this problem domain.

Molecular docking was chosen as a test problem as it constitutes a challenging real-life optimization problem of high importance in the fields of bioinformatics and biochemistry. The BNSO algorithm Bee-Nest was tested on the the PDBbind *core-set* [20] using ParaDocks [13], a docking framework developed for the application of population-based metaheuristics. The solutions obtained from Bee-Nest and the reference algorithms PSO, RNDM, and RHC were compared in terms of lowest energies, energy distribution, and RMSD to the reference solution.

With regards to the energy levels reached by optimizing with the BNSO, this algorithm's performance is very promising. In comparison to the three reference algorithms the BNSO algorithm Bee-Nest is able to generate receptor-ligand conformations with the lowest energy levels for the majority of the test instances. This trend is also reflected in the mean and median energy levels of the protein conformations the algorithms generated for the test instances.

RMSD values of the generated conformations constitute a measurement to estimate how close the produced conformations resemble the conformation found in wet lab experiments. Surprisingly, the RSMD values of the conformations produced by the BNSO are not as accurate as their energy levels would suggest. In terms of RSMD values, the tested PSO algorithm outperforms the BNSO. Around 46% of the conformations produced by the PSO are in a range of $2.5\mathring{A}$ to the real protein conformation, while this only holds for 36% of the conformations produced by the BNSO, even though many of them exhibit a lower energy value according to the scoring function. There are two potential explanations for this quality difference of the conformations produced by the Bee-Nest. One is that this might be due to the used scoring functions, thus tests with different scoring functions should be employed to see if the difference remains. Another potential explanation is that the BNSO algorithm Bee-Nest has problems overcoming the vast fitness barriers imposed by the molecular docking fitness landscapes. Lower energy values would then be the result of the Bee-Nest's superiority in fine tuning the protein conformations regarding its surrounding. If this is the case, a hybrid approach where the PSO is applied as a means of search space sampling and the BNSO algorithm functions as a post-processing algorithm might yield a very good performance if applied to molecular docking.

7. ACKNOWLEDGEMENTS

This work was supported by the Human Frontier Science Program Research Grant ”optimization in natural systems: ants, bees and slime moulds”. RM gratefully acknowledges financial support by SFB 610.

8. REFERENCES

[1] K. Diwold, M. Beekman, and M. Middendorf. Bee nest site selection as an optimization process. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (Alife XII)*, pages 626–633, Cambridge, MA, 2010. The MIT Press.

[2] K. Diwold, M. Beekman, and M. Middendorf. *Handbook of Swarm Intelligence - Concepts, Principles and Applications*, chapter Honeybee Optimisation. Springer, 2010.

[3] M. Farooq. *Bee-Inspired Protocol Engineering: From Nature to Networks*. Natural Computation Series. Springer, 2008.

[4] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Princliples of docking: an overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, and Genetics*, 47:409–443, 2002.

[5] S. Janson, D. Merkle, and M. Middendorf. Molecular docking with multi-objective particle swarm optimization. *Applied Soft Computing*, 8:666–675, 2008.

[6] S. Janson, M. Middendorf, and M. Beekman. Searching for a new home – scouting behavior of honeybee swarms. *Behavioral Ecology*, 18:384–392, 2007.

[7] G. Jones, P. Willett, and R. C. Glen. Molecular regcognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of Molecular Biology*, 245:43–53, 1995.

[8] P. Kampstra. Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software*, 28, 2008.

[9] D. Karaboga and B. Akay. A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31:61–85, 2009.

[10] O. Korb, T. Stützle, and T. Exner. Plants: Application of ant colony optimization to structure-based drug design. In *Proceedings of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence, ANTS 2006*, volume 4150 of *LNCS*, pages 247–258. Springer, 2006.

[11] B. Liu, H. Chen, H. Huang, S. Hwang, and S. Ho. Flexible protein-ligand docking using particle swarm optimization. In *IEEE Congress on Evolutionary Computation CEC2005*, volume 1, pages 251–258, 2005.

[12] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8:3–30, 1998.

[13] R. Meier, M. Pippel, F. Brandt, W. Sippl, and C. Baldauf. PARADOCKS: A Framework for Molecular Docking with Population-Based Metaheuristics. *Journal of Chemical Information and Modeling*, 50:879–889, 2010.

[14] I. Muegge. PMF scoring revisited. *Journal of Medical Chemistry*, 49(20):5895–5902, 2006.

[15] M. R. Myerscough. Dancing for a decision: a matrix model for nest-site choice by honeybees. *Proceedings of the Royal Society of London series B*, 270:577–582, 2003.

[16] T. Schmickl and K. Crailsheim. Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25:171–188, 2008.

[17] K. M. Schultz, K. M. Passino, and T. D. Seeley. The mechanism of flight guidance in honeybee swarms: subtle guides or streaker bees? *Journal of Experimental Biology*, 7:3287–3295, 2008.

[18] T. D. Seeley. *Honeybee Democracy*. Princeton University Press, Princeton, NJ, 2010.

[19] T. D. Seeley and P. K. Visscher. Sensory coding of nest-site value in honeybee swarms. *Journal of Experimental Biology*, 211:3691–3697, 2008.

[20] R. Wang, X. Fang, Y. Lu, and S. Wang. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medical Chemistry*, 47(12):2977–2980, 2004.