

Article

A Firefly-Inspired Method for Protein Structure Prediction in Lattice Models

Brian Maher ¹, Andreas A. Albrecht ², Martin Loomes ², Xin-She Yang ² and Kathleen Steinhöfel ^{1,*}

¹ Department of Informatics, King's College London, Strand, London WC2R 2LS, UK;
E-Mail: Brian.Maher@kcl.ac.uk

² School of Science and Technology, Middlesex University, The Burroughs, London, NW4 4BT, UK;
E-Mails: A.Albrecht@mdx.ac.uk (A.A.A.); M.Loomes@mdx.ac.uk (M.L.);
X.Yang@mdx.ac.uk (X.-S.Y.)

* Author to whom correspondence should be addressed; E-Mail: Kathleen.Steinhofel@kcl.ac.uk;
Tel.: +44-(0)20-7848-1590; Fax: +44-(0)20-7848-2851.

Received: 1 December 2013; in revised form: 17 December 2013 / Accepted: 27 December 2013 /
Published: 7 January 2014

Abstract: We introduce a Firefly-inspired algorithmic approach for protein structure prediction over two different lattice models in three-dimensional space. In particular, we consider three-dimensional cubic and three-dimensional face-centred-cubic (FCC) lattices. The underlying energy models are the Hydrophobic-Polar (H-P) model, the Miyazawa–Jernigan (M-J) model and a related matrix model. The implementation of our approach is tested on ten H-P benchmark problems of a length of 48 and ten M-J benchmark problems of a length ranging from 48 until 61. The key complexity parameter we investigate is the total number of objective function evaluations required to achieve the optimum energy values for the H-P model or competitive results in comparison to published values for the M-J model. For H-P instances and cubic lattices, where data for comparison are available, we obtain an average speed-up over eight instances of 2.1, leaving out two extreme values (otherwise, 8.8). For six M-J instances, data for comparison are available for cubic lattices and runs with a population size of 100, where, *a priori*, the minimum free energy is a termination criterion. The average speed-up over four instances is 1.2 (leaving out two extreme values, otherwise 1.1), which is achieved for a population size of only eight instances. The present study is a test case with initial results for *ad hoc* parameter settings,

with the aim of justifying future research on larger instances within lattice model settings, eventually leading to the ultimate goal of implementations for off-lattice models.

Keywords: protein folding; lattice models; H-P and M-J energy functions; Firefly Algorithm

1. Introduction

Research on protein structure and folding has a long history, dating back to the seminal work by Pauling and Corey [1]; see also [2–5]. From a computational point of view, all-atom protein structure prediction is a challenging task. Interestingly, Finkelstein and Badretdinov [6] approximated the worst case folding time of a protein of length n as $\exp(\lambda \cdot n^{2/3} \pm \chi \cdot n^{1/2}/2) ns$, where λ and χ are constants close to unity. In terms of algorithmic complexity, this implies a quasi-exponential structure prediction time with an exponent $\approx n^{2/3}$, if the structure prediction follows the folding path and elementary steps reflect structure transitions. Consequently, protein structure prediction has been shown to be NP-hard for various lattice models [7,8].

A common approach to tackle NP-hard problems are population-based methods and search-based heuristics, such as genetic algorithms [9], tabu search [10], Monte Carlo methods [11], simulated annealing [12–14] and quantum optimization [15,16]. Furthermore, simplified energy models are used for approximations of tertiary protein structures, with the most popular being the Hydrophobic-Polar (H-P) energy model [17–20] and the Miyazawa–Jernigan (M-J) energy model [21,22]. In the H-P model, proteins are represented by chains (within a given lattice) whose vertices are marked either as H (hydrophobic) or P (hydrophilic); H nodes are considered to attract each other, while P nodes are neutral. An optimal lattice embedding is one that maximises the number of H-H contacts, where the underlying justification is the assumption that hydrophobic interactions contribute to a significant portion of the total energy function. The M-J energy model is a more realistic model of the free energy of a given conformation, since it takes into account the interactions between specific pairs of amino acids.

In general, lattice models have been shown to be useful for the study of globular proteins [17,19], of the conformational space induced by (simplified) protein sequences [4,20] and for the analysis of a number of other features important in protein structure prediction, such as long-range interactions in proteins [23]. While typical lattice models account only for the backbone representation of protein sequences, recent research aims at incorporating the space required for side chain packing; see the LatFittool in [24]. As pointed out by Moreno-Hernández and Levitt in [20], protein structure predictions carried out in practice may consist of multiple stages that combine computational models of different complexities, where simplified models are often used in the early stages of structure predictions. In the present study, we focus on lattice models as a test case, with future research aiming at the application of the new search methodology to off-lattice models.

For comprehensive information about recent developments regarding lattice-based protein structure prediction, we refer the reader to [25,26]. In the centre of the present paper is the adaptation of a brand new population-based method to protein structure prediction, namely, the Firefly Algorithm (FA)

devised by X.S. Yang [27,28]. To the best of our knowledge, FA has been applied before to protein structure prediction only in [29] (the 2D H-P model).

For the H-P model and face-centred-cubic (FCC) lattices, a new tabu search heuristic is applied in [30] to 21 sequences of a length between 90 and 279. The new heuristic is able to improve on previously known minimum energy values in double digit percentages. The details of how to select favourable intermediate conformations in a population-based search are addressed in [31] (the H-P model and FCC lattices). The information can then be utilised for tabu search or related methodologies. Two energy models, namely the H-P model and a contact matrix similar to the M-J model, proposed in [32], are merged in [33,34]. The authors obtain improvements over existing methods in terms of the root mean square deviation (RMSD, the overall distance between the predicted structure and the native folding identified by X-ray crystallography or nuclear magnetic resonance spectroscopy) for a number of sequences out of 12 proteins with a length ranging from 54 to 160.

In [35], the authors devise a set of general guidelines for the application of Ant Colony Optimisation to protein structure prediction in lattice models. The guidelines are derived from computational experiments on 2D and 3D rectangular lattices combined with the H-P model.

In our study, we try to demonstrate the advantage of the Firefly-inspired method in terms of energy function evaluations, calculated over all individual runs of the population-based method. The results encourage us to analyse further larger instances for the H-P model and to adapt the approach to off-lattice models.

2. Experimental Section

2.1. Lattice Models

In order to simplify the modelling of protein conformations, and to make computational processing of the structure easier, the common practice is to represent the protein on a lattice. In informal terms, a lattice can be defined as any two- or three-dimensional graph. In our case, we will be using three-dimensional graphs, which are infinite in all three directions. A valid conformation on a lattice is a self-avoiding walk of the graph representation of the lattice, of a length equal to the length of the protein sequence being modelled. We will consider two lattice types in our experiments: the three-dimensional cubic lattice and the three-dimensional face-centred-cubic (FCC) lattice.

The formal definition of lattices is adopted from [26,36,37]: A lattice, Λ , is a subset of \mathbb{R}^N defined by:

$$\Lambda = \left\{ \mathbf{v} = \sum_{i=0}^{N-1} a_i \mathbf{u}_i \mid a_i \in \mathbb{Z} \right\}, \quad (1)$$

where the vectors, $\mathbf{u}_0, \dots, \mathbf{u}_{N-1}$, form a basis in \mathbb{R}^N and can be represented by the basis matrix $\mathbf{B}_\Lambda = (\mathbf{u}_0, \dots, \mathbf{u}_{N-1})$.

The three-dimensional cubic lattice is the simplest lattice we consider, where \mathbf{B}_Λ is given by:

$$\mathbf{B}_{3\text{Dcubic}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

Two nodes $\mathbf{v}_i = (x_i, y_i, z_i)$, $\mathbf{v}_j = (x_j, y_j, z_j) \in \mathbf{B}_{3\text{Dcubic}}$ are neighbours, i.e., connected by an edge, if:

$$|x_i - x_j| + |y_i - y_j| + |z_i - z_j| = 1. \quad (3)$$

Therefore, each node in the lattice has six neighbouring nodes, e.g., for $\mathbf{v} = (x, y, z)$, the neighbours are $(x + 1, y, z)$, $(x - 1, y, z)$, $(x, y + 1, z)$, $(x, y - 1, z)$, $(x, y, z + 1)$ and $(x, y, z - 1)$.

The FCC lattice is more complex and is a three-dimensional lattice formed around triangles. Each lattice node has 12 possible neighbouring nodes, twice that of the three-dimensional cubic lattice. We consider the FCC lattice, since, even though there are less published results for proteins folded over it, it has been shown that the FCC lattice provides good approximations of actual protein structures (see [14] for the relation to Kepler's conjecture). The basis matrix, \mathbf{B}_Λ , for the FCC lattice can be defined by:

$$\mathbf{B}_{\text{FCC}} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & \frac{\sqrt{6}}{3} \end{pmatrix}, \quad (4)$$

see [26] and also Section 3 in [36]. Two nodes $\mathbf{v}_i = (x_i, y_i, z_i)$, $\mathbf{v}_j = (x_j, y_j, z_j) \in \mathbf{B}_{\text{FCC}}$ are neighbours according to:

$$(|x_i - x_j| \leq 1) \& (|y_i - y_j| \leq 1) \& (|z_i - z_j| \leq 1) \& (|x_i - x_j| + |y_i - y_j| + |z_i - z_j| = 2). \quad (5)$$

For both of these lattice models, a contact between two amino acids is defined as any two amino acids on lattice nodes (x_i, y_i, z_i) and (x_j, y_j, z_j) that meet their respective neighbouring properties, defined above (Equations (3) & (5)), and that have the extra requirement that the two nodes do not hold P_n and P_{n+1} , where P_n represents the n -th amino acid in the protein sequence, P .

In our population-based model, we consider a separate infinite lattice for each conformation in the population. Conformations are represented as vectors, and as such, we can measure the distance between them over multiple lattices. Using separate lattices for each conformation prevents a move from not being made due to a node already containing an amino acid from another conformation.

All conformations in the population can move independently, which results in an algorithm that, to a certain degree, can be processed in parallel. For a given number of intermediate steps, T , the conformational moves from S_0 to S_T can be processed in parallel, for each conformation individually. It is only after each T step that we need all conformational moves to be completed, so that the current best result can be established to become part of the Firefly objective function.

2.2. Energy Functions

In a further simplification used to aid computational modelling, bioinformaticians consider simplified energy functions to determine the free energy of any valid conformation. We will consider two energy functions, the simplest, Hydrophobic-Polar (H-P) model proposed by Dill [18], and the Miyazawa–Jernigan (M-J) [21,22] energy model.

The H-P energy model is possibly the simplest energy model in general use. It considers each amino acid to be either hydrophobic or polar. The free energy is calculated as the negative sum of the number of non-sequential H-H bonds in the conformation. Due to its simplicity, the H-P model makes calculating

the free energy of a conformation fast, but it does not take into account the interactions between specific pairs of amino acids.

In more detail, let $\tilde{\alpha}_S$ denote a lattice embedding of the protein sequence, S . The embedding is *valid*, if $\tilde{\alpha}_S$ lies along a non-self-intersecting path within the lattice in such a way that adjacent vertices of the chain, S , occupy adjacent locations. We define the set of conformations by:

$$F_S := \{ \tilde{\alpha}_S \mid \tilde{\alpha}_S \text{ is a valid lattice embedding for } S \}. \quad (6)$$

Let HH_c denote the function that counts the number of neighbouring hydrophobic amino acids in $\tilde{\alpha}_S$ that are not neighbours in S , but are neighbours of a distance of one within the lattice. The objective function is then defined by:

$$E(\tilde{\alpha}_S) := -HH_c(\tilde{\alpha}_S). \quad (7)$$

The energy function corresponds to the notation *HP100* introduced in [20], which dates back to the energy function defined in [18]. Apart from *HP100*, Moreno-Hernández and Levitt [20] also consider *HP211*, where H-H contacts are weighted with -2 , and H-P, as well as P-P contacts are weighted with -1 .

The M-J energy model provides a more realistic model of the free energy of a given conformation, since it takes into account the interactions between specific pairs of amino acids. It is represented as a two-dimensional matrix of contact energies between pairs of amino acids; in our experiments, we will use the contact energies matrix presented in [21,22]. Whilst the M-J model is much more biologically realistic, it is more complicated to both implement and compute.

In the original paper [21], where the energy function is introduced, there are two different interaction matrices. The first matrix, often referred to as MJain the literature, stands for the actual energy value of each bond, while the second matrix, MJb, stands for the pairwise contributions to the total free energy related to the fact that two amino acids are forced to expel a solvent molecule and form a contact. In our study, we use the second matrix. The choice is driven by the fact that benchmarks with a provably optimal energy value in the cubic lattice are available for this matrix, MJb [38].

The total free energy under the M-J pairwise interactions is given by Equation (8), where $\{r_i\}$ is the set of bead coordinates that define $\tilde{\alpha}_S$. Let $AT(i)$ denote the amino acid type of the i -th residue of $\tilde{\alpha}_S$, and $e_{u,v}$ are the entries of the MJb matrix. We have, for the contact function $D(r_i - r_j) = 1$, if beads i and j form a contact (that is not a covalent linkage) within the lattice, and $D(r_i - r_j) = 0$, otherwise. We then define:

$$E(\tilde{\alpha}_S) = \sum_{i>j}^N e_{AT(i),AT(j)} D(r_i - r_j). \quad (8)$$

In the evaluation of our methods, we will consider different combinations of lattices and energy functions. We will present the following sets of results: The H-P model over the three-dimensional cubic lattice the H-P model over the three-dimensional FCC lattice, the M-J model over the three-dimensional cubic lattice and a modification of the M-J model introduced in [32] over the three-dimensional FCC lattice. We will use the number of energy evaluations as a performance benchmark of our algorithm.

2.3. Benchmark Sequences

2.3.1. H-P Benchmarks

The H-P benchmark problems listed in Table 1 are from [39,40] and represent standard structures in this area of research. The benchmark problems have been studied before by the authors [13,14]; see also [41,42].

Table 1. Hydrophobic-Polar (H-P) model benchmark problems (length of 48) from [39,40]. E_{bench} is the benchmark minimum on the 3D cubic lattice and E_{nat} the corresponding value for the face-centred-cubic (FCC) lattice.

ID	structure	E_{bench}	E_{nat}
HP1	hphhpphhhhphhhpphhpphhphhhphhphhpphhpppppppphh	-32	-69
HP2	hhhhphhphhhhhpphhpphhpppppppphhpppphhphhphhhph	-34	-69
HP3	phphhphhhhhpphhpphhphhphhphpppphhpphhpphhpphhp	-34	-72
HP4	phphhpphhhhpphhphhpphhhhhhpphhphhphpppphhpphhp	-33	-71
HP5	pppppphhhhhhpphhhhphhphhphhpphhpphhpppppphhphhph	-32	-70
HP6	hhpppphhphhphhphhphhphpppppphhpphhpppphhhhhhhhp	-32	-70
HP7	phpppphhhhhhphhphhphhphhpppphhpppphhhhpphhpphhp	-32	-70
HP8	phhphhhphhhhhpphhhhpppppphhpphhphhphpphhphhphpp	-31	-69
HP9	phphpppphhphhphhphhphhhhhhhpphhhhpphhpphhhhpppph	-34	-71
HP10	phhpppppphhpphhhhphpphhphhpphhpphhpphhpphhhhhhpphh	-33	-68

2.3.2. M-J Benchmarks

The M-J sequences taken from [38] are also of a length of 48; see Table 2.

Table 2. Miyazawa–Jernigan (M-J) model benchmark problems (length 48) from [38], with minimum energy results for 3D cubic lattices.

ID	structure	E_{nat}
MJ1	frtrplnhdfynykiwepfkpadfkwdrmlhdhvwdsmaswghqhc	-25.85
MJ2	cdlppftyrrhghndfwknyemikhwldlwrdfrafwsdpvkaspqas	-25.92
MJ3	frtpwvshqfyayklmehfkwdgfcnmkwidslpdrwnpaphdhas	-26.09
MJ4	kdkihfrmnygypawdaqsvkdltpdrwhfphmrdpshnwelaffws	-25.87
MJ5	endvtmdmdpspelfrihnlprahsfdrfgwhqfdkyhykwkwawaps	-26.15
MJ6	ehdaqlfdwsrwtwhgrnsyhapamyrwpvhdmkpnkfkiffles	-26.24

Along with the sequences from [38], we also consider four sequences analysed in [37] and [33] for 3D FCC lattices; see Table 3. However, the underlying energy matrix in [33] is taken from [32]. Therefore, we executed computational experiments for 3D FCC lattices on benchmarks from Table 3 for the energy matrix defined in [32].

Table 3. M-J Model benchmark problems for 3D FCC lattices from [37].

ID	structure	length
4RXN	mkkylctvcgyiydpedgdpddgvnpqgdfkdpiddwvpcplcgvgkdefeevee	54
1ENH	rpptafsseqlarkrefnenrylterrrqqlsselglnaqikiwfnkraki	54
4PTI	rpdfcleppytgpckariiryfynakaglcqtfvyggcrakrnnfksaedcmrtcgga	58
2IGD	mtpavtyklvingtklkgetttkavdaetaekafkqyandngvldgvwtyddatktftvte	61

2.4. Firefly Algorithm

Swarm intelligence (SI) and bio-inspired computation have received great interests and attention in the literature. SI-based algorithms, such as particle swarm optimization and the Firefly Algorithm, have simplicity, flexibility and lower time complexity, and thus, they can have many advantages over conventional algorithms [44,45].

2.4.1. Standard Firefly Algorithm

The Firefly Algorithm (FA) was developed by X.S. Yang in 2008 [27,28], which was based on the flashing patterns and behaviour of tropical fireflies. FA is simple, flexible and easy to implement. The basic assumptions in the Firefly Algorithm are:

- All fireflies are unisex, so that one firefly will be attracted to other fireflies regardless of their sex;
- Attractiveness is proportional to the their brightness; thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness, and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly by random walks;
- The brightness of a firefly is affected or determined by the landscape of the objective function.

As a firefly’s attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness, β , of a firefly by:

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{9}$$

where $\beta_0 \in (0, 1]$ is the attractiveness at $r = 0$.

The movement of a firefly, i , attracted to another more attractive (brighter) firefly, j , is determined by:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j^t - \mathbf{x}_i^t) + \alpha \boldsymbol{\epsilon}_i^t, \tag{10}$$

where the second term is due to the attraction. In addition, $\gamma > 0$ is a scaling factor and should be related to the scales of the problem of interest. The third term is randomization with α being the randomization parameter, and $\boldsymbol{\epsilon}_i^t$ is a vector of random numbers drawn from a Gaussian distribution at time t . Obviously, the randomization term, $\boldsymbol{\epsilon}_i^t$, can also be drawn from other probability distributions, such as Lévy flights [27,28]. A comprehensive review of the Firefly Algorithm and its variants has been carried out by Fister *et al.* [46].

For this algorithm to reach convergence properly, randomness should be gradually reduced, and one way to achieve such randomness reduction is to use:

$$\alpha = \alpha_0 \theta^t, \quad \theta \in (0, 1), \quad (11)$$

where t is the index of iterations/generations. Here, α_0 is the initial randomness factor, and we can set $\alpha_0 = 1$ without losing generality.

As for the parameter values, parametric studies [27,46] suggested that $\alpha_0 \in [0.5, 1]$, $\theta \in [0.7, 0.975]$, $n = 15$ to 100, and $\gamma = 0.01$ to 10, though $\alpha_0 = 1$ and $\gamma = 1$ can work for many applications.

From the above equation, we can see that mutation can be used for both local and global searches. When ϵ_i^t is drawn from a Gaussian distribution and Lévy flights, it produces a mutation on a larger scale. On the other hand, if α is chosen to be a very small value, then mutation can be very small and, thus, limited to a subspace. During the updating steps in the two loops in the FA, ranking, as well as selection is used, see Figure 1. A brief analysis of the firefly algorithm can reveal two distinct advantages:

- The automatic subdivision of the whole population into subgroups;
- The natural capability of dealing with multi-modal optimization.

All these advantages make FA very unique and very efficient.

2.4.2. Special Cases of FA

The Firefly Algorithm can have rich characteristics. First, it uses attraction to influence the behavior of the population. As local attraction tends to be stronger than long-distance attraction, the population in FA can automatically subdivide into subgroups, depending on the modality of the problem, which enables FA to deal with multimodal, nonlinear optimization problems naturally.

Furthermore, if we look at the updating Equation (10) more closely, we can see that there are special cases when some parameters are approaching certain limits. Firstly, if γ is very large, then attractiveness or light intensity decreases too quickly; this means that the second term in Equation (10) becomes negligible, leading to the standard simulated annealing (SA) [47]; see Section 2.5.2..

Secondly, if γ is very small (*i.e.*, $\gamma \rightarrow 0$), then the exponential factor $\exp[-\gamma r_{ij}^2] \rightarrow 1$. We have:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta_0(\mathbf{x}_j^t - \mathbf{x}_i^t) + \alpha \epsilon_i^t. \quad (12)$$

Here, if we further set $\alpha = 0$, then the above Equation (12) becomes a variant of a differential evolution [48]. On the other hand, if we replace \mathbf{x}_j^t by the current global best solution, \mathbf{g}^* , we have:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta_0(\mathbf{g}^* - \mathbf{x}_i^t) + \alpha \epsilon_i^t, \quad (13)$$

which is essentially a variant of particle swarm optimization [44] or, more specifically, the accelerated particle swarm optimization (APSO) introduced by Yang *et al.* [49].

Thirdly, if we set $\beta_0 = 0$ and let ϵ_i^t be related to \mathbf{x}_i , then Equation (12) becomes a pitch adjustment variant in harmony search (HS) [50].

Therefore, we can essentially say that DE, APSO, SA and HS are special cases of the Firefly Algorithm. Conversely, FA can be considered as a good combination of all four algorithms (DE, APSO,

SA and HS), to a certain extent. Furthermore, FA uses a nonlinear updating equation, which can produce richer behaviour and higher convergence than the linear updating equations used in standard PSO and DE. Consequently, it is no surprise that FA can outperform other algorithms in many applications, such as multimodal optimization, classifications, image processing and feature selection [46].

Figure 1. Pseudocode of the Firefly Algorithm (FA).

```

Objective function  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ .
Generate an initial population of  $n$  fireflies  $\mathbf{x}_i$  ( $i = 1, 2, \dots, n$ ).
Light intensity  $I_i$  at  $\mathbf{x}_i$  is determined by  $f(\mathbf{x}_i)$ .
Define light absorption coefficient  $\gamma$ .
while ( $t < \text{MaxGeneration}$ ),
for  $i = 1 : n$  all  $n$  fireflies
    for  $j = 1 : n$  all  $n$  fireflies (inner loop)
        if ( $I_i < I_j$ )
            Move firefly  $i$  towards  $j$ .
        end if
        Vary attractiveness with distance  $r$  via  $\exp[-\gamma r^2]$ .
        Evaluate new solutions and update light intensity.
    end for  $j$ 
end for  $i$ 
Rank the fireflies and find the current global best  $\mathbf{g}_*$ .
end while
Post-process results and visualization.

```

2.5. Firefly-Inspired Protein Structure Prediction

Whilst our approach does not follow the exact ideals of the Firefly methodology, it could be considered a simulated annealing-Firefly hybrid method. We apply the principles of the Firefly optimisation algorithm to a more traditional population-based simulated annealing algorithm, with a primary goal of minimising the number of objective function evaluations (defined as $Oval_{min}$ and $Oval_{avg}$ in our results tables) required to reach a solution.

2.5.1. Pull Move Set

The algorithm we have developed uses the pull moves set [10] with the correction suggested by Györfy *et al.* in [51] (see p. 1848, left column) for cubic lattices. We choose the (adjusted) pull move set, because it has been shown to be reversible and complete for cubic lattices [10,51] and FCC lattices [36] (Section 3), meaning that every possible valid conformation is reachable using only the pull move set. We use this pull move set in conjunction with a local search approach, combined with simulated annealing, in order to find an optimum embedding in the lattice. This approach is used together with the Firefly methodology, as shown below, over both the three-dimensional cubic and FCC lattices.

2.5.2. Simulated Annealing

Simulated annealing has long been used [47] as an optimisation method for local search-based problems. We implement simulated annealing to our local search method to avoid becoming trapped in a local minimum during the early stages of computation. Becoming trapped in a local minimum could prevent an optimal embedding from being reached, since the move to an optimal conformation may not be possible if intermediate conformations with a less-optimal energy function value are not considered.

Since the objective of this experiment is to determine the effectiveness of the Firefly optimisation method, we consider only a linear cooling schedule and not a logarithmic cooling schedule.

We use a standard cooling schedule in accordance with:

$$T_n = 4000 \times 0.9999^n, \quad (14)$$

and a simulated annealing decision based on:

$$e^{-\frac{|E(S')-E(S)|}{T_n}}. \quad (15)$$

This function is used to determine whether a move is to be taken. The value of n is not increased for each member of the population, that is, after 10 steps in a population of a size of 5, $n = 10$, not 50. This ensures that population members who are computed sequentially after another have the same probability of accepting a move at any given step.

2.5.3. Population

Our method is a population-based method. We consider a population of a size of $N = 8$ in our experiments, where this number is at the lower end of the settings for N discussed in population-based computing (e.g., if values of $N = 2^n = 4, 8, 16, 32, \dots$ are considered). Moreover, due to the multi-threaded nature of our implementation on four CPU cores, the communication overhead is relatively small for $N = 8$.

We create our population by identifying a set of initial conformations. To do this for a population of a size of N , we create N infinite lattice models. We then make a random self-avoiding walk of each member lattice, of the same length as the protein sequence, attaching the value of each amino acid in the sequence to its respective node. Once we have this random initial conformation, we run a greedy version of the pull move set algorithm on each initial conformation, until we reach a local minimum. This greedy version of the pull move set will only accept moves from the pull move set that result in a lower (better) energy function evaluation.

In traditional simulated annealing combined with population-based methods, the entire population is considered for simulated annealing-based evaluation using the energy function at every step. A crucial feature of our approach is that simulated annealing-based decisions are executed at each step only for the conformation with the lowest value of the energy function (at the last known iteration of T steps), whereas the remaining members of the population are evaluated at each step according to their Euclidean distance to the best conformation. The selection of the best conformation is then updated after T steps, as described in Section 2.5.4..

2.5.4. Integration with Firefly

We use the Firefly method as a tool for guiding intermediate solutions towards optimum solutions, where the (intermediate) best solution within the population serves as a reference structure for a limited number of T steps and changes during this interval according to simulated annealing decisions.

We consider the initial population ($S_1 \dots S_N$) and an interval, T . At the beginning, pull moves are executed T times with the only criterion that the energy value does not increase (greedy descent, potentially to local minima within T steps). We then perform an energy evaluation of all conformations in the generated initial population, and save the conformation with the lowest energy value as S_{best} (random selection, if there are several conformations with the lowest energy value according to Equation (7) or Equation (8)). It is important to note that the member, S_{best} , of the population remains the same member during simulated annealing-guided conformational changes during each interval of T steps. In terms of the standard Firefly Algorithm, S_{best} , can be considered to be the firefly with the brightest light (and, thus, the light intensity of a population member is inversely proportional to its energy function evaluation; that is, lower energy function evaluations result in brighter firefly lights).

While for each T -interval, standard pull moves are executed and evaluated based upon simulated annealing on S_{best} , for every other conformation in the population ($N - 1$), pull moves are applied with respect to an alternative objective function: In order to mimic the Firefly method, each of these $N - 1$ conformations is evaluated in accordance with its Euclidean distance to S_{best} , i.e., the second objective function favours changes to the corresponding lattice embedding that move the conformation structurally towards the current S_{best} .

In formal terms, the structural similarity of [S_{best}, S_i] is defined by the Euclidean distance: For the 3D coordinates (x^t, y^t, z^t) of the $t - th$ node (amino acid) at step n within the T -interval, we set:

$$d_n^t(i) = \sqrt{(x_{best}^t - x_i^t)^2 + (y_{best}^t - y_i^t)^2 + (z_{best}^t - z_i^t)^2}. \quad (16)$$

If, now, a pull move affects the positions of m nodes (amino acids) from position j until position k ($m = k - j + 1$), we select the average value of the distance changes as the structural objective function:

$$\text{sof}_{n+1}(S_i) = \frac{1}{m} \sum_{t=j}^k |d_n^t - d_{n+1}^t|. \quad (17)$$

The pull move is only accepted for the population member, S_i , if $\text{sof}_n(S_i) \geq \text{sof}_{n+1}(S_i)$. Thus, if the total distance increases, but more nodes are affected, the move could still be accepted, which can contribute to the diversity of the population.

Our approach follows the Firefly optimisation principle, since all members of the population with dimmer lights (higher energy function evaluations) move (structurally) to the member with the brightest light (the lowest energy evaluation score).

After T steps, we perform an energy function evaluation on all N members of the population, and the member of the population that is labelled S_{best} is updated.

The procedure terminates if there are no pull moves accepted by simulated annealing for the conformation labelled as S_{best} within the current T -interval, and the member of the population labelled as S_{best} remains the same after this particular interval.

Figure 2. Pseudocode of the integration of the Firefly Algorithm and local search.

Objective function $sa(S_{1...N})$ executes pull move based on simulated annealing
 Objective function $struct(S_{1...N})$ executes pull move based on the distance to S_{best}
 Energy evaluation function $eval(S_{1...N})$ calculates the energy value of the conformation

Define $T \leftarrow 15$ (pre-defined interval)
 Define $iterations \leftarrow 0$ (current iteration count)

Generate N conformations $S_{1...N}$ (random self-avoiding walks)

for $i = 1 : N$ all N conformations
 while (S_i has pull-move m , which results in reduced energy)
 Perform move m on S_i
 end while
end for i

$S_{best} \leftarrow S_x$, where S_x is the conformation with the best energy function evaluation
 $best_index \leftarrow x$

while running
 $flagged \leftarrow False$
 for $i = 1 : N$ all N conformations
 if $i = best_index$
 Execute pull move on S_i based on the simulated annealing ($sa()$) objective function
 else
 Execute pull move on S_i based on the structural move ($struct()$) objective function
 end if
 if $eval(S_i) < eval(S_{best})$
 $flagged \leftarrow True$
 end for i
 $iterations \leftarrow iterations + 1$
 if $iterations = T$ **or** $flagged = True$
 $S_{best} \leftarrow S_x$, where S_x is the conformation with the best energy function evaluation
 $best_index \leftarrow x$
 $iterations \leftarrow 0$
 end if
end while

As an overview (see the pseudocode in Figure 2), the main steps of the algorithm are:

1. Generate a random initial population of random walks (conformations).
2. Perform pull moves on each conformation greedily until a local minimum is found.
3. Copy the best conformation in the population into S_{best} . Note its index in $best_index$.
4. Repeat steps 5 through 7.
5. Perform a pull move on conformation $best_index$ using the simulated annealing objective function.
6. Perform a pull move on all other conformations using the structural objective function.
7. If we have reached T iterations since S_{best} , we set a new value for S_{best} and $best_index$ by picking randomly from all conformations with the lowest energy function evaluation.

3. Results and Discussion

In our experiments, we tested the performance of our algorithm on both the H-P and M-J energy functions on three-dimensional cubic and FCC lattices. We take the minimum and average results and objective function counts from five runs of each combination. All experiments are run with a population size of eight and a default interval of $T = 15$. By selecting such a small value of T , we try to avoid that conformations different from S_{best} do not move for too many steps structurally towards a conformation that is no longer the best with respect to the energy value. Future research will include finding out whether or not one can identify a range of T (depending on the sequence length) more appropriate for this particular parameter setting.

The computational experiments were executed on computers with Intel Xeon E5-4603 CPUs and 16 GB of RAM, running Debian 6 x64. The runtimes for the HP1–HP10 sequences ranged from 35 to 142 minutes on the 3D cubic lattice and from 81 to 250 minutes on the FCC lattice. For sequences MJ1–MJ5 over the cubic lattice, runtimes ranging from 89 to 221 minutes were observed. We expect these runtimes to improve with further fine-tuning of the algorithm and parameter settings.

The performance of algorithms is difficult to compare solely based upon runtime values, since the particular runtimes are affected by the underlying hardware and implementation details. Therefore, we provide data about the total number of objective function evaluations, which is denoted by *Oval*. The notation has been chosen in order to cover both energy function evaluations according to Equation (7) and Equation (8), which are counted for the member of the population labelled as S_{best} and the number of ‘structure-based’ objective function evaluations according to Equation (17), which are executed for each of the remaining $N - 1$ members of the population. Thus, if $K = k \times T + T'$ is the number of steps (without the initial stage) until the termination of the procedure, where $1 \leq T' \leq T$, then:

$$\text{Oval} = \sum_{i=1}^N \text{Init}_i + N(k \times T + T') + (k + 1)(N - 1). \quad (18)$$

The values of Init_i cover the initial stage before the selection of S_{best} , and $(k + 1)(N - 1)$ accounts for the selection of S_{best} at the beginning of each complete T -interval and T' (we note that the energy value

of S_{best} is known at the beginning of each interval). By selecting such a performance measure, it is also possible to circumvent the runtime impact of how the underlying lattices (which are equal for the same problem setting) are implemented and handled in computational experiments.

In the literature, comparable data with respect to function evaluations are provided in [52] with respect to the H-P model (ten benchmarks from Table 1), and in [41] for the M-J model (six benchmarks from Table 2); in both cases, for the three-dimensional cubic lattice. The function evaluations are counted for the respective energy function, and therefore, we denote by Eval the total number of energy function evaluations. We note that in [52], the value of Eval is equal to the product of the population size (= 20) and the number of ‘generations’. Since the number of ‘generations’ reported in [52] is the average value over 10 independent runs, we use $Eval_{avg}$ for comparison purposes. In [41] (see Tables 9-14 there), 10 independent runs are reported individually with their respective Eval value and minimum energy value. The population size for each of the runs is 100. For comparison purposes, we calculated the average $Eval_{avg}$ value over 10 runs for each of the six M-J instances.

In our computational experiments, we executed five independent runs, and we denote by $Oval_{avg}$ the average number of objective function evaluations according to Equation (18) for each individual benchmark problem. For comparison to [41] and [52], we introduce the parameter:

$$speed-up = \frac{Eval_{avg}}{Oval_{avg}} \tag{19}$$

Table 4 displays our results using the H-P model on the three-dimensional cubic lattice. It shows that, over the 10 benchmark problems, our method finds an optimal conformation for every problem. This is comparable to other methods; as E_{pub} shows, the method presented in [52] achieved the same performance.

Table 4. Results for the H-P model on 3D cubic lattices for benchmarks from Table 1. $Eval_{avg}^{pub}$ values are from the results published in [52].

ID	E_{bench}	E_{pub}	Z_{min}	Z_{avg}	$Eval_{avg}^{pub} \times 10^6$	$Oval_{min} \times 10^6$	$Oval_{avg} \times 10^6$	$Oval_{max} \times 10^6$	speed-up
HP1	-32	-32	-32	-29.8	7.43	11.28	17.53	24.62	0.4
HP2	-34	-34	-34	-33.4	74.57	7.87	16.02	26.16	4.7
HP3	-34	-34	-34	-32.2	19.19	6.61	12.65	22.97	1.5
HP4	-33	-33	-33	-30.4	12.03	8.95	14.39	23.96	0.8
HP5	-32	-32	-32	-31.2	12.78	4.58	10.33	16.69	1.2
HP6	-32	-32	-32	-30.8	10.63	12.15	17.42	19.08	0.6
HP7	-32	-32	-32	-30.6	917.37	6.35	12.99	15.84	70.6
HP8	-31	-31	-31	-29.0	15.20	6.34	12.98	26.54	1.2
HP9	-34	-34	-34	-32.6	23.95	9.35	15.94	24.62	1.5
HP10	-33	-33	-33	-32.0	42.46	3.88	7.80	17.24	5.4
average speed-up total									8.8
average speed-up, leaving out HP1 and HP7									2.1

Except for HP1, HP4 and HP6, our method improves on the number of energy function evaluations reported in [52], and on eight out of the ten instances, $Oval_{\min}$ is smaller than $Eval_{\text{avg}}^{\text{pub}}$. If the worst (HP1) and best (HP7) performances are left out, we achieve an average speed-up of 2.1.

A potential explanation for the relatively large value of $Eval_{\text{avg}}^{\text{pub}}$ for HP7 could be the landscape analysis carried out in [14] for this instance: HP7 exhibits the smallest γ value, which results in a termination criterion larger by about one margin compared to most of the other sequences (see Table 1 in [14]).

Table 5 shows our results for the H-P model over the FCC lattice. This combination shows similar performance to the three-dimensional cubic lattice, in that both our method and the published [53] method found an optimal conformation in 100% of the benchmark problems. We can also see that in 50% of benchmarks (HP2, HP3, HP4, HP5 and HP8), we achieved a better average conformation value than the results published in [53].

Table 5. Results for the H-P model on 3D FCC lattices for benchmarks from Table 1. E_{pubMin} and E_{pubAvg} are the published minimum and average results from [53]. Published energy function evaluation counts are not available.

ID	E_{nat}	E_{pubMin}	E_{pubAvg}	Z_{\min}	Z_{avg}	$Oval_{\min}$ $\times 10^6$	$Oval_{\text{avg}}$ $\times 10^6$	$Oval_{\max}$ $\times 10^6$
HP1	-69	-69	-67.37	-69	-66.2	18.98	23.57	29.74
HP2	-69	-69	-66.97	-69	-67.8	25.08	47.97	69.68
HP3	-72	-72	-68.80	-72	-69.8	23.69	32.22	39.77
HP4	-71	-71	-68.10	-71	-69.4	15.89	21.70	27.46
HP5	-70	-70	-67.77	-70	-68.0	18.70	31.70	38.92
HP6	-70	-70	-66.93	-70	-66.2	13.81	23.80	28.37
HP7	-70	-70	-67.57	-70	-65.0	25.59	37.51	42.22
HP8	-69	-69	-66.37	-69	-68.4	21.11	33.52	41.38
HP9	-71	-71	-69.10	-71	-68.8	16.17	23.12	29.40
HP10	-68	-68	-66.47	-68	-65.6	32.14	45.22	53.72

To our knowledge, there are no published energy function evaluation counts for the benchmark problems HP1–HP10 over the 3D FCC lattice. Comparing to our results for the three-dimensional cubic lattice, there is an increase in energy evaluations over all benchmark problems. This is to be expected considering the additional complexity of the 3D FCC lattice.

Table 6 shows our results from five runs for the M-J model on the three-dimensional cubic lattice. In this combination, we achieved an optimal conformation in four out of the six benchmarks. We recall that the $Eval_{\text{avg}}^{\text{pub}}$ values are the average values of energy function evaluation counts over all 10 runs obtained by PLS in [41] for the corresponding benchmark. We note that except for MJ3, not all of the 10 runs reported in [41] reached the optimum value (ranges from six up to nine runs on the remaining instances).

On four of the benchmarks, we achieved the optimum free energy value. With the *ad hoc* parameter settings we used, we missed out by a margin on MJ1 and MJ4. It should be noted that the speed-up of objective function evaluations in comparison to PLS from [41] is 1.5 and 1.4 on MJ1 and MJ4,

respectively. Therefore, we think it is justified to say that, with a slightly higher computational effort, our approach will be able to achieve the optimum free energy values on these two instances. When leaving out MJ3 and MJ5, the average speed-up is 1.2 (otherwise, 1.1). As mentioned before, the population size in [41] is 100, and we presume that with a moderate increase of the population size for the Firefly-inspired algorithms, a stronger speed-up can be achieved. Furthermore, it is important to note that in [41], the runs terminate when a structure with minimum free energy has been found, which is not the case in our computational experiments.

Table 6. Results for the M-J Model on 3D cubic lattices for benchmarks from Table 2. E_{pub} are the results published in [41]. $\text{Eval}_{\text{avg}}^{\text{pub}}$ are the average energy function evaluation counts for PLS from [41].

ID	E_{nat}	E_{pub}	Z_{min}	Z_{avg}	$\text{Eval}_{\text{avg}}^{\text{pub}} \times 10^6$	$\text{Oval}_{\text{min}} \times 10^6$	$\text{Oval}_{\text{avg}} \times 10^6$	$\text{Oval}_{\text{max}} \times 10^6$	speed-up
MJ1	-25.85	-25.85	-24.18	-19.58	33.01	15.24	21.56	28.12	1.5
MJ2	-25.92	-25.92	-25.92	-24.16	12.81	17.37	28.33	33.87	0.5
MJ3	-26.09	-26.09	-26.09	-24.92	6.02	14.76	23.79	27.35	0.3
MJ4	-25.87	-25.87	-23.59	-18.30	34.58	18.54	24.59	23.66	1.4
MJ5	-26.15	-26.15	-26.15	-25.01	23.02	7.91	13.20	19.09	1.7
MJ6	-26.24	-26.24	-26.24	-22.73	36.32	13.73	25.96	36.80	1.4
average speed-up total									1.1
average speed-up, leaving out MJ3 and MJ5									1.2

Table 7 shows our results obtained by using the energy function from [32] over the FCC lattice compared to those from [33].

Table 7. Results for the energy pairwise interactions energy function from [32] on 3D FCC lattice for the benchmarks from Table 3. E_{pub} are the published results from [33].

ID	length	E_{pub}	Z_{min}	Z_{avg}	$\text{Oval}_{\text{min}} \times 10^6$	$\text{Oval}_{\text{avg}} \times 10^6$	$\text{Oval}_{\text{max}} \times 10^6$
4RXN	54	-166.88	-166.21	-158.90	20.05	39.82	52.78
1ENH	54	-153.79	-150.64	-144.12	16.86	35.09	47.52
4PTI	58	-210.29	-213.52	-200.86	23.98	31.88	58.56
2IGD	61	-183.18	-185.15	-179.88	34.71	64.87	111.22

Our results are comparable to [33] on instance 4RXN, worse on instance 1ENH and better on the longer instances, 4PTI and 2IGD. Unfortunately, no data are available in published literature for comparison of the number of energy function evaluations.

4. Conclusions

We presented the initial results of protein structure prediction in lattice models by using simplified energy function and a new Firefly-inspired heuristic with *ad hoc* parameter settings. With regard to H-P instances and the minimum number of objective function evaluations, our methods outperform the published results on eight out of ten benchmark problems for cubic lattices. If the average number of objective function evaluations over five runs is compared to the published data over ten runs, the average speed-up is 2.1 over eight instances, where two extreme cases are left out (0.4 and 70.6). For H-P instances on FCC lattices, all ten optimum free energy values are achieved with a relatively small number of objective function evaluations. For the six M-J instances of a length of 48, we obtained on cubic lattices an average speed-up over four instances of 1.2, leaving out two extreme values (otherwise, 1.1). The speed-up is achieved for a population size of only eight instances, whereas the population size producing the data we used for comparison is 100. We presume the speed-up can be improved, if a larger population size is chosen for our Firefly-inspired method. On four selected benchmark problems of a length between 54 and 61, we achieved for the energy matrix proposed in [32] on FCC lattices an improvement of minimum free energy values on two instances in comparison to the results published in [33]. As in the case of H-P instances on FCC lattices, the number of energy function evaluations is comparably small. The results we obtained can be seen as a proof of concept, which encourages us to move forward towards the analysis of longer benchmark problems and off-lattice models.

Acknowledgements

The authors are grateful to the anonymous referees for their careful reading of the manuscript and valuable suggestions, which was very supportive for producing an improved and more detailed presentation of the results.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Pauling, L.; Corey, R.B. Stable configurations of polypeptide chains. *Proc. Roy. Soc. London B* **1953**, *141*, 21–33.
2. Scheraga, H.A. *Protein Structure*; Academic Press: New York, NY, USA, 1961.
3. Levitt, M.; Warshel, A. Computer simulation of protein folding. *Nature* **1975**, *253*, 694–698.
4. Hinds, D.A.; Levitt, M. Exploring conformational space with a simple lattice model for protein structure. *J. Mol. Biol.* **1994**, *243*, 668–682.
5. Sali, A.; Shakhnovich, E.; Karplus, M. How does a protein fold? *Nature* **1994**, *369*, 248–251.
6. Finkelstein, A.V.; Badretdinov, A.Y. Rate of protein folding near the point of thermodynamic equilibrium between the coil and the most stable chain fold. *Fold. Des.* **1997**, *2*, 115–121.
7. Paterson, M; Przytycka, T. On the complexity of string folding. *Discrete Appl. Math.* **1996**, *71*, 217–230.

8. Berger, B.; Leighton, T. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *J. Comput. Biol.* **1998**, *5*, 27–40.
9. Unger, R.; Moulton, J. Genetic algorithms for protein folding simulations. *J. Mol. Biol.* **1993**, *231*, 75–81.
10. Lesh, N.; Mitzenmacher, M.; Whitesides, S. A complete and effective move set for simplified protein folding. In Proceedings 7th Annual International Conference on Research in Computational Molecular Biology, Berlin, Germany, April 2003; Vingron, M., Istrail, S., Pevzner, P., Waterman, M., Eds.; ACM: New York, NY, USA, 2003; pp. 188–195.
11. Zhang, J.-F.; Kou, S.C.; Liu, J.S. Biopolymer structure simulation and optimization via fragment regrowth Monte Carlo. *J. Chem. Phys.* **2007**, *126*, 225101.
12. Hansmann, U.H.E.; Okamoto, Y. Monte Carlo simulations in generalized ensemble: Multicanonical algorithm versus simulated tempering. *Phys. Rev. E* **1996**, *54*, 5863–5865.
13. Steinhöfel, K.; Skaliotis, A.; Albrecht, A.A. Relating time complexity of protein folding simulation to approximations of folding time. *Comp. Phys. Comm.* **2007**, *176*, 165–170.
14. Albrecht, A.A.; Skaliotis, A.; Steinhöfel, K. Stochastic protein folding simulation in the three-dimensional HP-model. *Comput. Biol. Chem.* **2008**, *32*, 248–255.
15. Perdomo, A.; Truncik, C.; Tubert-Brohman, I.; Rose, G.; Aspuru-Guzik, A. Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models. *Phys. Rev. A* **2008**, *78*, 012320-1–012320-15.
16. A. Perdomo-Ortiz, A.; N. Dickson, N.; M. Drew-Brook, M.; G. Rose, G.; and A. Aspuru-Guzik, A. Finding low-energy conformations of lattice protein models by quantum annealing. *Sci. Rep.* **2012**, *2*, 571.
17. Dill, K.A. Theory for the folding and stability of globular proteins. *Biochemistry* **1985**, *24*, 1501–1509.
18. Lau, F.F.; Dill, K.A. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules* **1989**, *22*, 3986–3997.
19. Dill, K.A.; Bromberg, S.; Yue, K.; Fiebig, K.M.; Yee, D.P.; Thomas, P.D.; Chan, H.S. Principles of protein folding—A perspective from simple exact models. *Protein Sci.* **1995**, *4*, 561–602.
20. Moreno-Hernández, S.; Levitt, M. Comparative modeling and protein-like features of hydrophobicpolar models on a two-dimensional lattice. *Proteins* **2012**, *80*, 1683–1683.
21. Miyazawa, S.; Jernigan, R.L. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules* **1985**, *18*, 534–552.
22. Miyazawa, S.; Jernigan, R.L. Residue–residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *J. Mol. Biol.* **1996**, *256*, 623–644.
23. Noivirt-Brik, O.; Unger, R.; Horovitz, A. Analysing the origin of long-range interactions in proteins using lattice models. *BMC Struct. Biol.* **2009**, *9*, 4.
24. Mann, M.; Saunders, R.; Cameron Smith, C.; Backofen, R.; Deane C.M. Producing high-accuracy lattice models from protein atomic coordinates including side chains. *Adv. Bioinf.* **2012**, *2012*, 148045.

25. Hart, W.E.; Newman, A. Protein structure prediction with lattice models. In *Handbook of Molecular Biology*; Aluru, S., Ed.; Computer and Information Science Series; Chapman & Hall/CRC: Boca Raton, FL, USA, 2006; pp. 30/1–30/24.
26. Bechini, A. On the characterization and software implementation of general protein lattice models. *PLoS ONE* **2013**, *8*, e59504.
27. Yang, X.S. Firefly algorithms for multimodal optimization. In Proceedings 5th International Symposium on Stochastic Algorithms: Foundations and Applications, Sapporo, Japan, October 2009; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin and Heidelberg, Germany, 2009; pp. 169–178.
28. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-inspired Comput.* **2010**, *2*, 78–84.
29. Zhang, Y.D.; Wu, L.; Wang, S.H. Solving two-dimensional HP model by firefly algorithm and simplified energy function. *Math. Prob. Engg.* **2013**, *2013*, 398141.
30. Rashid, M.A.; Newton, M.A.H.; Hoque, M.T.; Shatabda, S.; Pham, D.N.; Sattar, A. Spiral search: a hydrophobic-core directed local search for simplified PSP on 3D FCC lattice. *BMC Bioinf.* **2013**, *14*(Suppl. 2), S16.
31. Shatabda, S.; Newton, M.A.H.; Rashid, M.A.; Pham, D.N.; Sattar, A. The road not taken: retreat and diverge in local search for simplified protein structure prediction. *BMC Bioinf.* **2013**, *14*(Suppl. 2), S19.
32. Berrera, M.; Molinari, H.; Fogolari, F. Amino acid empirical contact energy definitions for fold recognition in the space of contact maps. *BMC Bioinf.* **2003**, *4*, 8.
33. Rashid, M.A.; Newton, M.A.H.; Hoque, M.T.; Sattar, A. Mixing energy models in genetic algorithms for on-lattice protein structure prediction. *BioMed Res. Intl.* **2013**, 924137.
34. Shatabda, S.; Newton, M.A.H.; Sattar, A. Mixed heuristic local search for protein structure prediction. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Washington, DC, USA, July 2013; desJardins, M., Littman, M.L., Eds.; AAAI Press: Palo Alto, CA, USA, 2013; pp. 876–882.
35. Nardelli, M.; Tedesco, L.; Bechini, A. Cross-lattice behavior of general ACO folding for proteins in the HP model. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, March 2013; Shin, S.Y., Maldonado, J.C., Eds.; 2013. ACM: New York, NY, USA, 2013; pp. 1320–1327.
36. Böckenhauer, H.; Ullah, A.Z.M.D.; Kapsokalivas, L.; Steinhöfel, K. A local move set for protein folding in triangular lattice models. In Proceedings Workshop on Algorithms in Bioinformatics, Karlsruhe, Germany, September 2008; Crandall, K.A., Lagergren, J., Eds.; Springer: Berlin and Heidelberg, Germany, 2008; pp. 369–381.
37. Ullah, A.D.; Steinhöfel, K. A hybrid approach to protein folding problem integrating constraint programming with local search. *BMC Bioinf.* **2010**, *11*(Suppl. 1), S39.
38. Faisca, P.F.N.; Plaxco, K.W. Cooperativity and the origins of rapid, single-exponential kinetics in protein folding. *Protein Sci.* **2006**, *15*, 1608–1618.
39. Yue, K.; Fiebig, K.M.; Thomas, P.D.; Chan, H.D.; Shakhnovich, E.I.; Dill, K.A. A test of lattice protein folding algorithms. *Proc. Natl. Acad. Sci. USA* **1995**, *92*, 325–329.

40. Beutler, T.C.; Dill, K.A. A fast conformational search strategy for finding low energy structures of model proteins. *Protein Sci.* **1996**, *5*, 2037–2043.
41. Kapsokalivas, K.; Gan, X.; Albrecht, A.A.; Steinhöfel, K. Population-based local search for protein folding simulation in the MJ energy model and cubic lattices. *Comput. Biol. Chem.* **2009**, *33*, 283–294.
42. Dotu, I.; Cebriá, M.; Van Hentenryck, P.; Clote, P. On lattice protein structure prediction revisited. *IEEE/ACM T. Comput. Biol. Bioinf.* **2011**, *8*, 1620–1632.
43. Plaxco, K.W.; Simons, K.T.; Baker, D. Contact order, transition state placement and the refolding rates of single domain proteins. *J. Mol. Biol.* **1998**, *277*, 985–994.
44. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings IEEE International Conference on Neural Networks, Perth, Australia, December 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 1942–1948.
45. Yang, X.S.; Hosseini, S.S.S.; Gandomi, A.H. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Appl. Soft Comput.* **2012**, *12*, 1180–1186.
46. Fister, I.; Fister, I., Jr.; Yang, X.S.; Brest, J. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.* **2013**, in press.
47. Kirkpatrick, S.; Gellat, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680.
48. Storn, R.; Price, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **1997**, *11*, 341–359.
49. Yang, X.S.; Deb, S.; Fong, S. Accelerated particle swarm optimization and support vector machine for business optimization and applications. In Proceedings Third International Conference on Networked Digital Technologies, Macau, China, July 2011; Fong, S., Ed.; Springer: Berlin and Heidelberg, Germany, 2011; pp. 53–66.
50. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: harmony search. *Simulation* **2001** *76*, 60–68.
51. Györfy, D.; Závodszy, P.; Szilágyi, A. “Pull moves” for rectangular lattice polymer models are not fully reversible. *IEEE/ACM T. Comput. Biol. Bioinf.* **2012**, *9*, 1847–1849.
52. Kapsokalivas, L.; Gan, X.; Albrecht, A.; Steinhöfel, K. Two local search methods for protein folding simulation in the HP and the MJ lattice models. In Proceedings Bioinformatics Research and Development, Vienna, Austria, July 2008; Elloumi, M., Küng, J., Linial, M., Murphy, R.F., Schneider, K., Toma, C., Eds.; Springer: Berlin and Heidelberg, Germany, 2008; pp. 167–179.
53. Su, S.; Tsay, J. Rotation crossover and K-site move mutation for evolutionary protein folding in 3D FCC HP model. In Proceedings IEEE International Conference on Bioinformatics and Biomedicine, Philadelphia, PA, USA, October 2012; Gao, J., Alhajj, R., Dubitzky, W., Ungar L., Wu, C., Christianson, A., Liebman, M., Hu, X., Eds.; IEEE: Piscataway, NJ, USA, 2012; pp. 1–4.