

# A Hybrid Monte Carlo Ant Colony Optimization Approach for Protein Structure Prediction in the HP Model

Andrea G. Citrolo    Giancarlo Mauri

Dipartimento di Informatica Sistemistica e Comunicazione  
Università degli Studi di Milano-Bicocca.

{citrolo,mauri}@disco.unimib.it

The hydrophobic-polar model has been widely studied in the field of protein structure prediction both for theoretical purposes and as a benchmark for new optimization strategies. In this work we introduce a new heuristics based on Ant Colony Optimization and Markov Chain Monte Carlo that we called Hybrid Monte Carlo Ant Colony Optimization. We describe this method and compare results obtained on well known HP instances in the 3-dimensional cubic lattice to those obtained with standard Ant Colony optimization and Simulated Annealing. All methods were implemented using an unconstrained neighborhood and a modified objective function to prevent the creation of overlapping walks. Results show that our methods perform better than the other heuristics in all benchmark instances.

## 1 Introduction

Proteins are polymers composed of linear chains of amino-acids. These molecules adopt a complex three-dimensional structure that allows them to take part in almost every biological process. Since proteins function strongly depends on their three-dimensional structure the availability of fast and reliable techniques for protein structure determination is of great interest in fields such as drug design, proteomics and synthetic biology. Nevertheless, experimental techniques for protein structure determination, like X-ray crystallography and nuclear magnetic resonance are still very expensive and time consuming. With the increasing number of sequence data available, the lack of a reliable method for protein structure prediction (PSP) risks to become the true bottleneck in the post-genomic era, making PSP one of the most challenging problems in computational biology. According to the widely accepted thermodynamic hypothesis [3], at least for small globular proteins, the native structure is the structure with the lowest potential energy and so the PSP can be treated as a global minimization problem. Real instances of PSP contain thousand of atoms independently moving in the three-dimensional space, so they are very hard to treat both in terms of computational resources and in terms of theoretical formalization. For these reasons some simplified models have been introduced in order to investigate fundamental aspects of the folding process. Among them the most studied is the hydrophobic-polar (HP) model introduced by Dill et al. [4] in which proteins are represented as strings in a binary alphabet; the goal is to find the optimal on-lattice self-avoiding walk for the given binary sequence. Each protein residue in the HP model is represented by a single position in the input string. The objective function counts the contacts between non-adjacent hydrophobic positions in the sequence. This model has been studied in a variety of lattices, and it has been proved that the optimization problem of maximizing the number of H-H contacts is NP-hard for a broad class of objective functions independently of the lattice chosen to define the discrete protein-folding model [6]. In recent years a great number of methods have been applied to the PSP in the HP model, ranging from constraint programming to heuristic approaches [2, 7, 8, 12, 14, 15, 16, 17, 19, 20].

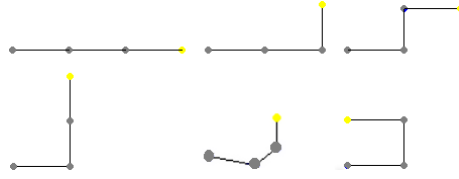


Figure 1: Allowed starting blocks

Analyzing these studies two considerations can be made: primary methods based on perturbation of complete solutions (like Replica Exchange Monte Carlo) seem to perform better than methods based on constructive approaches (with the notable exception of the pruned-enriched Rosenbluth method). In addition methods that keep memory of good solutions to prevent re-sampling performs better than methods that do not [16]. In this work we introduce a new heuristics that combines Ant Colony Optimization (ACO) with the Monte Carlo method in order to overcome major drawbacks of these techniques.

## 2 Solution Representation

In a three-dimensional cubic lattice residue positions are encoded using three integer numbers representing respectively values along x, y and z Cartesian axis. In the literature solutions coding for a generic HP string  $s$  of length  $n$  are generally represented as strings of length  $n - 1$ , in which each position can assume one of six values indicating an unitary movement along one of the six directions. The first position is arbitrarily placed at the origin of the axes. In this work we use a slightly different solution representation that is similar to the one adopted in off-lattice models based on the fragment assembly strategy [22]. We build 150 fragments that cover all the valid conformations of an HP string of length four. Each fragment determines the local structure of a triplet of residues starting at the origin and also the position of the first residue in the following triplet. To prevent the creation of rotational duplicates only the six blocks in fig .1 are allowed at the first position.

## 3 Objective Function

The canonical objective function of the HP problem considers only the number of contacts between non consecutive hydrophobic positions (H-H contacts) and the problem is set as a global minimization. As we will discuss later, in this work we use a simple neighborhood structure based on blocks exchange. This Neighborhood does not guarantee that generated solutions are self-avoiding walks. In order to discourage the generation of overlapping walks and to improve the performance of the constructive step we introduced the modified scoring function shown in eq. 1.

$$Energy = -\left[\left(\frac{HHscore}{1 + OVscore}\right)^2 + \frac{1}{1 + HPscore} + \frac{2tHH}{1 + HDscore}\right] \quad (1)$$

where  $tHH$  is the upper bound of number of contacts defined in [9]. The general form of a *score* is defined above:

$$score = \sum_{i=1}^{n-3} \sum_{j=i+3}^n f(\chi_i, \chi_j, s_i, s_j) \quad i, j \in \mathbb{N} \quad (2)$$

where  $\chi_i, \chi_j \in \mathbb{Z}^3$  are the vectors of coordinates of residue  $i, j$  respectively while  $s_i, s_j \in [0, 1]$  are values of sequence at position  $i, j$  (we arbitrarily choose 1 for hydrophobic positions and 0 for polar positions),  $f$  is a function that defines the specific type of score. The *HHscore* represents the canonical objective function of the HP problem and is obtained setting  $f = \kappa$  as defined in in eq. 3 :

$$\kappa(\chi_i, \chi_j, s_i, s_j) = \begin{cases} -1 & \text{if } \sum_{k=1}^3 |\chi_{i,k} - \chi_{j,k}| = 1 \wedge s_i = s_j = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

*OVscore* uses  $f = \omega$  shown in eq. 4; it counts the number of overlapping positions. To also consider the overlaps at the boundary between blocks the  $\omega$  function is also computed between residues  $i$  and  $i + 1$  at the boundary positions.

$$\omega(\chi_i, \chi_j, s_i, s_j) = \begin{cases} 1 & \text{if } \sum_{k=1}^3 |\chi_{i,k} - \chi_{j,k}| = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

*HPscore* uses  $f = \phi$  shown in eq. 5 ; it counts the contacts between hydrophobic and polar positions (H-P contacts), it was introduced to prevent the formation of undesired H-P contacts; a similar term was used in [7].

$$\phi(\chi_i, \chi_j, s_i, s_j) = \begin{cases} 1 & \text{if } \sum_{k=1}^3 |\chi_{i,k} - \chi_{j,k}| = 1 \wedge s_i \neq s_j \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

finally the *HDscore* uses  $f = \delta$  in eq. 6 and it was introduced to bias the early steps of constructive methods toward compact solutions.

$$\delta(\chi_i, \chi_j, s_i, s_j) = \begin{cases} \sum_{k=1}^3 |\chi_{i,k} - \chi_{j,k}| & \text{if } i, j \text{ have different parity } \wedge s_i = s_j = 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

## 4 Graph mapping

Since both ACO heuristics and Hybrid Monte Carlo Ant Colony Optimization (HMCACO) require an explicit probabilistic model that influences transitions between states of the modeled system a graph mapping of PSP problem is needed. Given an HP string  $s$  of length  $n$ , we define  $b_s \in \mathbb{N}$  to be the size of a structural block in terms of sequence coverage ( $b_s = 3$  in our representation),  $n_b \in \mathbb{N}$  to be the number of structural blocks used to represent each sub-sequence of size  $b_s$  ( $n_b = 150$ ). Furthermore we define:

$$ID = \{i \in \mathbb{N} | i \leq n/b_s\}, \quad LS = \{m \text{ is a valid local structure of size } b_s\}, \quad A = ID \times LS \quad (7)$$

an undirected graph  $G$  of size  $(n_b \frac{n}{b_s})$  such that there is a one to one correspondence between nodes of  $G$  and elements of  $A$ . In the general case the set of edges of  $G$  contains one edge between any pair of node in  $i, j$  for which holds  $id_i \neq id_j$  in the complementary representation. Finally, we define  $G'$  that contains the same node set of  $G$  but with edges connecting only pairs of nodes for which it holds  $|id_i - id_j| = 1$  and oriented in the direction of increasing values of  $id$ . The HP problem can be formalized as a search for paths of size  $\frac{n}{b_s}$  in a dynamical graph dynamical system (GDS) defined on  $G'$  in which the state of the system is obtained through a backtracking function that builds the coordinates matrix corresponding to the structure encoded by the nodes of the path. Since each path on  $G'$  generates a state that is a valid argument for the evaluation function in eq .1, the PSP problem is that of finding the lowest energy path.

As we will see, both ACO and HMCACO use the  $G$  graph to keep track of good pairwise relations between structural blocks obtained by lower energy paths sampled by ants in the DGS defined on  $G'$ . This information is then used to bias the following samplings of the GDS.

## 5 Methods

### 5.1 Hill Climbing

This simple iterative heuristics is based on the concept of neighborhood. The neighborhood is built through the definition of a neighborhood function of the following type:

Let  $I$  be an instance of an optimization problem

Let  $X$  be the set of all feasible solutions for  $I$

Let  $H_s$  be the set of neighbor solutions of  $s$ ,

$$s, \in X, H_s \subseteq X$$

we define:

$$N : X \rightarrow 2^X : \quad \forall s \in X \rightarrow N(s) \in H_s$$

In this work we state that a solution  $a$  is  $k$ -neighbor of a solution  $b$  if and only if they differs for less than  $k$  structural block. The hill climbing procedure used in this work performs at each iteration an exhaustive search over the 1-neighborhood of the input solution. The best neighbor found according to eq .1 is thus selected as the new current solution only if its cost is lower than the cost of the input solution. This heuristics was used as local optimizer in both ACO and HMCACO.

### 5.2 Simulated Annealing

Here we give just an idea of the basic concept underlying the Simulated Annealing method, for an extensive treatment of this topic see [10, 21]. Simulated Annealing (SA) is one of the most studied neighborhood based heuristics of the broad Markov Chain Monte Carlo (MCMC) family of methods. Many heuristics in this family have proved to perform very well on HP instances when the neighborhood is obtained through pool moves [11] or with the move set described in [5]. A general Markov Chain Monte Carlo method works in an iterative fashion sampling the search space according to a given distribution. The Boltzmann distribution is the most used one for polymer simulation. This means that most of the sampling time is spent on solutions with low energy values. Each iteration of a MCMC method can be divided into two phases: sampling, and transition. In the sampling phase a random neighbor of the current solution is generated, then in the acceptance phase the Metropolis Hastings acceptance criterion in eq .8 is applied, and the result is compared to a random number between  $[0,1)$ . If  $p$  is greater than the random number, the neighbor solution is accepted as starting solution of the next iteration.

$$p(s, s') = \begin{cases} 1 & \text{if } c'_s \leq c_s \\ \exp \frac{c_s - c'_s}{T} & \text{otherwise} \end{cases} \quad (8)$$

Here  $p(s, s')$  is the probability to make the transition from solution  $s$  to neighbor solution  $s'$ , while  $c_i$  is the value of the objective function of solution  $i$ . The peculiarity of SA is that the temperature varies during the search process. At the beginning of the search high values of temperature are used to facilitate a broad exploration of the search space, after each iteration the temperature is decreased using a problem-dependent cooling scheme. In this work we used for SA the same neighborhood function presented for Hill Climbing and we based the cooling on the scheme presented in [2].

### 5.3 Ant Colony Optimization

Ant colony optimization is a bio-inspired meta-heuristics to approach hard combinatorial problems in which a colony of simple agents (artificial ants) interact to efficiently explore the search space. The only applicability condition to satisfy in order to use ACO is the availability of a graph mapping for the transitions in the target problem[13]. As we explained before, transitions of the PSP problem can be mapped on a graph. The general idea of ant inspired systems is that of combining the constructive strategy with a global evaluation stochastic heuristics and to keep memory of the relations between solution components in high quality solutions. This relation is stored in a matrix called pheromone matrix that is a real number representation of the edges of the previously described transition graph. In this work we adopted the *Max-Min* Ant System meta-heuristics [18]; this implies that the quantity of pheromone in each position of the pheromone matrix is bounded by a minimal and a maximal value. Each iteration of the meta-heuristics is composed of three main steps: construction, evaluation and daemon actions. The construction step works like a probabilistic greedy algorithm. A starting node with  $id = 1$  is chosen probabilistically according to the mean of pheromone values associated to its edges (in case of PSP only nodes with  $id = 1$  are considered as candidate starting nodes). All the subsequent nodes that will enter the solution are chosen in probability as follows:

$$p(i, j) = \frac{t_{ij}^{\alpha} \eta^{\beta}}{\sum_{b=1}^{n_b} t_{ib}^{\alpha} \eta_c^{\beta}} \quad (9)$$

Here  $p(i, j)$  is the probability of including node  $j$  in the growing solution,  $t_{ij}$  is the pheromone value on the  $ij$ -edge  $\eta$  is the cost of extending the current partial solution with node  $j$ , computed using a heuristic function,  $\alpha$  and  $\beta$  are algorithm parameters. Since ants build solutions in a constructive fashion, the pheromone matrix for ACO uses only edge between consecutive block in the sequence so in this case the edge set of  $G$  coincides with that of  $G'$ . In the daemon-step, solutions that have been built from the ants undergo local optimization. Then the pheromone level is decreased for all the edges according to eq .10.

$$\tilde{t}_{ij} = (1 - \rho)t_{ij} + \rho t_{min} \quad (10)$$

Here  $\tilde{t}_{ij}$  is the pheromone value on  $ij$ -edge after evaporation, while  $\rho$  is a parameter. In the evaluation step the best solution of the current iteration is compared to the best solution obtained so far. If current solution is an improvement over the best solution, the latter is updated and then used to increase the pheromone level on the contained edges, as shown in eq. 11. Otherwise, the releaser is chosen using some heuristic criterion:

$$\hat{t}_{ij} = \begin{cases} \tilde{t}_{ij} + \frac{c_s}{c_{opt}^*} & \text{if } \tilde{t}_{ij} < t_{max} \\ t_{max} & \text{otherwise} \end{cases} \quad (11)$$

Here  $\hat{t}_{ij}$  is the value of pheromone matrix at positions  $i, j$  after the release, while  $c_s$  is the cost of the releaser solution and  $c_{opt}^*$  is an estimate of the optimal value for the given instance. At the beginning of the algorithm the whole pheromone matrix is set to the max pheromone value. In this work we used five ants for each iteration; we set  $\alpha = \beta = 1$ ,  $\rho = 0.1$ . The policy to set and to update values of  $t_M$  and  $t_m$  has been defined following suggestions from [13].

ID	sequence
S1	hphhpphhhhphhhpphphpphphhhphphhphpphpppppppph
S2	hhhhphhphhhhhpphphpphphpppppphpphpphphhphhhph
S3	phhphhhhhhhpphphpphphhphpphpphphhphhphpphp
S4	phhphpphphhhpphphpphphhhhhpphphhphpppphpphph
S5	pphppphhhhhpphhhhphhphhhpphphpphpppppphphhph
S6	hhppphhphhphhphhphpppppphphpphpphphhhhhhhph
S7	phpppphphhhphhhhhphhphpphphpphphhphpphphpph
S8	phhphhhphhhhhpphhpppppphphhphhphpphphhphhppp
S9	phhpppphphpphphhhhhpphhhhphpphphhphhphhpppph
S10	phhpppppphpphphhphpphphhpphpphphhphhphhphh

Table 1: Standard benchmark sequences for 3D HP problem in cubic lattice

## 5.4 Hybrid Monte Carlo Ant Colony Optimization

As anticipated in the introduction, the technique we present here is aimed to combine the pheromone biased search typical of ACO with the perturbation approach of a Markov Chain Monte Carlo method. The general structure of the algorithm is that of an Ant based heuristics already described. The main difference with ACO is that in the construction step each ant is initialized to a model solution, and a set of pheromone based perturbations is applied. This introduces the concept of neighborhood typical of a perturbation based approach. During perturbation each candidate node is evaluated using the heuristic information and the mean of the pheromone values computed over all edges connecting the incoming node to nodes in the solution that are not going to be replaced. The probability to select a generic incoming node thus becomes:

$$p(i, j) = \frac{(\frac{1}{|S|} \sum_{\sigma=1}^{|S|} t_{ij}^{\alpha}) \eta^{\beta}}{\sum_{b=1}^{n_b} (\frac{1}{|S|} \sum_{\sigma=1}^{|S|} t_{\sigma b}^{\alpha}) \eta_c^{\beta}} \quad (12)$$

Here  $p(i, j)$  is the probability to accept node  $j$  as perturbing node at position  $i$ ;  $S$  is the set containing nodes in the ant that are not going to be replaced by node  $j$ , so  $|S| = \frac{n}{b_s} - 1$ , and  $\sigma$  is an index over  $S$ . It is clear from eq .12 that all the edges of  $G$  are considered in the pheromone matrix of HMCACO. A second modification introduced in HMCACO with respect to the standard ACO is that comparison between best so far solution and iteration best solution is made using the Metropolis Hastings criterion eq .8. The accepted solution is allowed both to release the pheromone on contained edges and to become the model for the next iteration. The values of the parameters for HMCACO are the same described for ACO; the optimal value for neighborhood size was found to be four.

## 6 Results and Discussion

In this work we chose the standard benchmark set for HP model in the three-dimensional cubic lattice taken from [1] and shown in tab.1 .

This set has been extensively used in other works [2, 12, 17, 19]. All the sequences in the benchmark set have length 48, and the global optimum for each of them has been obtained using the CPSP tool [14]. This program uses a constraint based approach to perform an exhaustive search for sequences of moderate size. All the methods have been implemented using the same library and tested on the same hardware, so

ID	$E_{min}$	SA	ACO	HMCACO
S1	-32	<b>-26</b> (-23.7)	<b>-29</b> (-27.0)	<b>-31</b> (-28.6)
S2	-34	<b>-27</b> (-23.9)	<b>-29</b> (-26.7)	<b>-32</b> (-29.2)
S3	-34	<b>-28</b> (-25.1)	<b>-28</b> (-26.7)	<b>-32</b> (-29.5)
S4	-33	<b>-28</b> (-24.1)	<b>-29</b> (-26.6)	<b>-31</b> (-29.1)
S5	-32	<b>-28</b> (-25.1)	<b>-29</b> (-26.5)	<b>-31</b> (-28.5)
S6	-32	<b>-24</b> (-23.0)	<b>-28</b> (-25.7)	<b>-30</b> (-27.7)
S7	-32	<b>-26</b> (-23.6)	<b>-28</b> (-26.5)	<b>-31</b> (-28.4)
S8	-31	<b>-27</b> (-24.1)	<b>-29</b> (-25.9)	<b>-29</b> (-27.8)
S9	-34	<b>-28</b> (-25.2)	<b>-30</b> (-27.8)	<b>-32</b> (-29.7)
S10	-33	<b>-27</b> (-24.3)	<b>-29</b> (-27.0)	<b>-31</b> (-29.6)

Table 2: Results of different heuristics compared to optimal values of energy for each benchmark sequence

we based the comparison on results obtained from short run of comparable length (CPU time  $< 2'$ ). In table tab.2 we show the best and average results for each instance computed over 50 runs for the different heuristics. None of the methods was able to reach the global optimum in the considered execution time. This is probably due to the availability of overlapping solutions introduced from the naive neighborhood structure we used. It is very likely that the presence of these solution alters the fitness landscape of the problem making it rougher. An argumentation in support of this supposition is that in the work of Albrecht et al. [2] SA with a neighborhood based on pull moves is able to reach the global optimum for several instances in a number of iterations lower than what we used in this work. In the case of ACO, previously reported results [17] indicate that also with other neighborhood structures the time required to reach the global optimum is longer than the time used in this work. The poor performances obtained from SA in this context however are not completely unexpected, if we consider the difference in size between a pull move based neighborhood and the one adopted in this work. Albrecht et al. in their study of time complexity of HP model for SA had shown that the number of iterations required to reach the global minimum in canonical HP structure prediction is bounded to:

$$t_{opt} = \left(\frac{m}{\delta}\right)^{D/\gamma} \quad (13)$$

Here  $t_{opt}$  is the number of iterations required to reach the global optimum of the problem with a confidence of  $\delta$ ,  $m$  is the average size of the neighborhood of a solution,  $D$  is the depth in terms of energy of the deepest local minimum for the given instance and  $\gamma$  is the average energy variation in an improving iteration. Even if it is hard to compare convergence properties of different models, we can assume that the size of the neighborhood has similar effects on the general behavior of SA, and this could explain the bad performance we observed. In the work of Albrecht, in fact, the value of  $m$  was estimated to be  $m \approx n/2$ , while in this work it is equal to the number of blocks allowed in each position  $m = 150$ . Another interesting observation is that ACO in this context performs better than SA. This is probably due to the pheromone bias that helps the method to avoid overlapping solutions and to spend more time on feasible solutions. This is true also for HMCACO that is able to sample low energy regions in the search space with increased efficacy. In our opinion this is an effect of the model-based perturbation strategy, since in standard ACO the constructive approach, even in the case of a single unlucky insertion, can push the search to regions of space with low pheromone content, impairing also the following insertion steps.

## 7 Conclusions and Future Works

In this work we presented a new heuristics based on ACO and Markov Chain Monte Carlo that we called HMCACO; we tested it on standard benchmarks of PSP in 3D-HP model with a naive neighborhood and a modified objective function. Results showed that in this context HMCACO outperforms both ACO and SA. Preliminary analysis of SA simulations indicate that neighborhood introduced in this work might not be well suited for the SA heuristics. This is interesting since for many aspects, the representation used here is closer to off-lattice model than the standard HP representation. Future work will be dedicated to establishing the efficacy of HMCACO for PSP both in HP models with the standard neighborhood and objective function and also in off-lattice models.

## References

- [1] *HPbenchmarks*. Available at [http://www.cs.sandia.gov/tech\\_reports/compbio/tortilla-hp-benchmarks.html](http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html).
- [2] a a Albrecht, a Skaliotis & K Steinhöfel (2008): *Stochastic protein folding simulation in the three-dimensional HP-model*. *Computational biology and chemistry* 32(4), pp. 248–55, doi:10.1016/j.compbiolchem.2008.03.004. Available at <http://www.ncbi.nlm.nih.gov/pubmed/18485827>.
- [3] CB Anfinsen (1973): *Principles that govern the folding of protein chains*. *Science* 181(4096), pp. 223–230. Available at [http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://web.iitd.ac.in/~amittal/1973\\_Anfinsen\\_Science.pdf](http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://web.iitd.ac.in/~amittal/1973_Anfinsen_Science.pdf). doi:10.1126/science.181.4096.223
- [4] K a Dill (1985): *Theory for the folding and stability of globular proteins*. *Biochemistry* 24(6), pp. 1501–9. Available at <http://www.ncbi.nlm.nih.gov/pubmed/3986190>. doi:10.1021/bi00327a032
- [5] Mehmet T. Gurler, Charles C. Crabb, Deborah M. Dahlin & Jeffrey Kovac (1983): *Effect of bead movement rules on the relaxation of cubic lattice models of polymer chains*. *Macromolecules* 16(3), pp. 398–403, doi:10.1021/ma00237a012. Available at <http://pubs.acs.org/doi/abs/10.1021/ma00237a012>.
- [6] WE Hart & S Istrail (1997): *Robust proofs of NP-hardness for protein folding: General lattices and energy potentials*. *Journal of Computational Biology* 4(1), pp. 1–22. Available at <http://online.liebertpub.com/doi/abs/10.1089/cmb.1997.4.1>. doi:10.1089/cmb.1997.4.1
- [7] HP Hsu, Vishal Mehra & Peter Grassberger (2003): *Growth algorithms for lattice heteropolymers at low temperatures*. *The Journal of chemical ...* Available at <http://link.aip.org/link/?JCPA6/118/444/1>.
- [8] Hsiao-Ping Hsu & Peter Grassberger (2011): *A Review of Monte Carlo Simulations of Polymers with PERM*. *Journal of Statistical Physics* 144(3), pp. 597–637, doi:10.1007/s10955-011-0268-x. Available at <http://www.springerlink.com/index/10.1007/s10955-011-0268-x>.
- [9] Sorin Istrail & Fumei Lam (2009): *Combinatorial algorithms for protein folding in lattice models: A survey of mathematical results*. *Communications in Information and Systems*, pp. 1–40. Available at <http://www.cs.brown.edu/people/sorin/pdfs/pfoldingsurvey.pdf>.
- [10] S Kirkpatrick, DG Jr. & MP Vecchi (1983): *Optimization by simulated annealing*. *science* 220(4598), pp. 671–680. Available at <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://home.gwu.edu/~stroud/classics/KirkpatrickGelattVecchi83.pdf>. doi:10.1126/science.220.4598.671



- [11] Neal Lesh, Michael Mitzenmacher & Sue Whitesides (2003): *A complete and effective move set for simplified protein folding*. *Proceedings of the seventh annual international conference on Computational molecular biology - RECOMB '03*, pp. 188–195, doi:10.1145/640075.640099. Available at <http://portal.acm.org/citation.cfm?doid=640075.640099>.
- [12] Faming Liang & Wing Hung Wong (2001): *Evolutionary Monte Carlo for protein folding simulations*. *The Journal of Chemical Physics* 115(7), p. 3374, doi:10.1063/1.1387478. Available at <http://link.aip.org/link/JCPSA6/v115/i7/p3374/s1&Agg=doi>.
- [13] Dorigo M. & Stützle T. (2004): *Ant Colony Optimization*. MIT Press. Available at <http://mitpress.mit.edu/books/ant-colony-optimization>.
- [14] Martin Mann, Sebastian Will & Rolf Backofen (2008): *CPSP-tools—exact and complete algorithms for high-throughput 3D lattice protein studies*. *BMC bioinformatics* 9, p. 230, doi:10.1186/1471-2105-9-230. Available at <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2396640&tool=pmcentrez&rendertype=abstract>.
- [15] A Piccolboni & G Mauri (1998): *Application of evolutionary algorithms to protein folding prediction*. In: *Lecture Notes in Computer Science*, 1363, pp. 123–135. Available at <http://link.springer.com/chapter/10.1007/BFb0026595>.
- [16] Swakkhar Shatabda, M a Hakim Newton, Mahmood a Rashid, Duc Nghia Pham & Abdul Sattar (2013): *The road not taken: retreat and diverge in local search for simplified protein structure prediction*. *BMC bioinformatics* 14 Suppl 2(Suppl 2), p. S19, doi:10.1186/1471-2105-14-S2-S19. Available at <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3549842&tool=pmcentrez&rendertype=abstract>.
- [17] Alena Shmygelska & Holger H Hoos (2005): *An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem*. *BMC bioinformatics* 6, p. 30, doi:10.1186/1471-2105-6-30. Available at <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=555464&tool=pmcentrez&rendertype=abstract>.
- [18] Thomas Stützle & Holger H. Hoos (2000): *MaxMin Ant System*. *Future Generation Computer Systems* 16(8), pp. 889–914, doi:10.1016/S0167-739X(00)00043-1. Available at <http://linkinghub.elsevier.com/retrieve/pii/S0167739X00000431>.
- [19] Chris Thachuk, Alena Shmygelska & Holger H Hoos (2007): *A replica exchange Monte Carlo algorithm for protein folding in the HP model*. *BMC bioinformatics* 8, p. 342, doi:10.1186/1471-2105-8-342. Available at <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2071922&tool=pmcentrez&rendertype=abstract>.
- [20] Abu Dayem Ullah & Kathleen Steinhöfel (2010): *A hybrid approach to protein folding problem integrating constraint programming with local search*. *BMC bioinformatics* 11 Suppl 1, p. S39, doi:10.1186/1471-2105-11-S1-S39. Available at <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3009511&tool=pmcentrez&rendertype=abstract>.
- [21] V Černý (1985): *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*. *Journal of optimization theory and applications* 45(1), pp. 41–51. Available at <http://link.springer.com/article/10.1007/BF00940812>. doi:10.1007/BF00940812
- [22] Dong Xu & Yang Zhang (2012): *Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field*. *Proteins: Structure, Function, and Bioinformatics* 80(November 2011), pp. 1715–1735, doi:10.1002/prot.24065. Available at <http://doi.wiley.com/10.1002/prot.24065>.