An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem

Alena Shmygelska, Rosalía Aguirre-Hernández, and Holger H Hoos*

Department of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
{oshmygel,rosalia,hoos}@cs.ubc.ca
WWW home page: http://www.cs.ubc.ca/labs/beta

Abstract. The prediction of a protein's conformation from its amino-acid sequence is one of the most prominent problems in computational biology. Here, we focus on a widely studied abstraction of this problem, the two dimensional hydrophobic-polar (2D HP) protein folding problem. We introduce an ant colony optimisation algorithm for this NP-hard combinatorial problem and demonstrate its ability to solve standard benchmark instances. Furthermore, we empirically study the impact of various algorithmic features and parameter settings on the performance of our algorithm. To our best knowledge, this is the first application of ACO to this highly relevant problem from bioinformatics; yet, the performance of our ACO algorithm closely approaches that of specialised, state-of-the methods for 2D HP protein folding.

1 Introduction

Ant Colony Optimisation (ACO) is a population-based approach to solving combinatorial optimisation problems that is inspired by the foraging behaviour of ant colonies. The fundamental approach underlying ACO is an iterative process in which a population of simple agents ("ants") repeatedly construct candidate solutions. This construction process is probabilistically guided by heuristic information on the given problem instance as well as by a shared memory containing experience gathered by the ants in previous iterations ("pheromone trails"). Following the seminal work by Dorigo *et al.* [3], ACO algorithms have been successfully applied to a broad range of hard combinatorial problems (see, *e.g.*, [4, 5]).

In this paper, we present an ACO algorithm for solving an abstract variant of one of the most challenging problems in computational biology: the prediction of a protein's structure from its amino-acid sequence. Genomic and proteomic sequence information is now readily available for an increasing number of organisms, and genetic engineering methods for producing proteins are well developed. The biological function and properties of proteins, however, are crucially determined by their structure. Hence, the ability to reliably and efficiently predict

* Corresponding author

protein structure from sequence information would greatly simplify the tasks of interpreting the data collected by the Human Genome Project, of understanding the mechanism of hereditary and infectious diseases, of designing drugs with specific therapeutic properties, and of growing biological polymers with the specific material properties.

Currently, protein structures are primarily determined by techniques such as MRI (magnetic resonance imaging) and X-ray crystallography, which are expensive in terms of equipment, computation and time. Additionally, they require isolation, purification and crystallisation of the target protein. Computational approaches to protein structure prediction are therefore very attractive. Many researchers view the protein structure prediction problem as the "Holy Grail" of computational biology; while considerable progress has been made in developing algorithms for this problem, the performance of state-of-the-art techniques is still regarded as unsatisfactory.

The difficulty in solving protein structure prediction problems stems from two major sources: (1) finding good measures for the quality of candidate structures (*e.g.*, energy models), and (2), given such measures, determining optimal or close-to-optimal structures for a given amino-acid sequence. The first of these issues needs to be addressed primarily by biochemists who study and model protein folding processes; the second, however, is a rich source of interesting and challenging computational problems in local and global optimisation. In order to separate these two aspects of protein structure prediction problems, the optimisation problem is often studied for simplified models of protein folding. In this work, we focus on the 2-dimensional hydrophobic-polar (2D HP) model, an extremely simple model of protein structure that has been used extensively to study algorithmic approaches to the protein structure prediction problem. Even in this simplified model, finding optimal folds is computationally hard (NP-hard) and heuristic optimisation methods, such as ACO, appear to be the most promising approach for solving this problem.

The remainder of this paper is structured as follows. In Section 2, we introduce the 2D HP model of protein structure, and give a formal definition of the 2D HP protein folding problem as well as a brief overview of existing approaches for solving this problem. Our new ACO algorithm for the 2D HP protein folding problem is described in Section 3. An empirical study of our algorithm's performance and the role of various algorithmic features is presented in Section 4. In the final Section 5 we draw some conclusions and point out several directions for future research.

2 The 2D HP Protein Folding Problem

The hydrophobic-polar model (HP model) of protein structure was first proposed by Dill [9]. It is motivated by a number of well-known facts about the pivotal role of hydrophobic and polar amino-acids for protein structure [9, 13]:

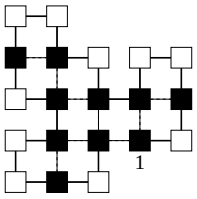

Fig. 1. A sample protein conformation in the 2D HP model. The underlying protein sequence (Sequence 1 from Table 1) is HPHPPHHPHPPHPHHPPHPH; black squares represent hydrophobic amino-acids while white squares symbolise polar amino-acids. The dotted lines represents the H-H contacts underlying the energy calculation. The energy of this conformation is -9, which is optimal for the given sequence.

- Hydrophobic interaction is the driving force for protein folding and the hydrophobicity of amino acids is the main force for development of a native conformation of small globular proteins.
- Native structures of many proteins are compact and have well-packed cores that are highly enriched in hydrophobic residues as well as minimal solvent-exposed non-polar surface areas.

Each of the twenty commonly found amino-acids that are the building blocks of all natural proteins can be classified as hydrophobic (H) or polar (P). Based on this classification, in the HP model, the primary amino-acid sequence of a protein (which can be represented as a string over a twenty-letter alphabet) is abstracted to a sequence of hydrophobic (H) and polar (P) residues, *i.e.*, amino-acid components. The conformations of this sequence, *i.e.*, the structures into which it can fold, are restricted to self-avoiding paths on a lattice; for the 2D HP model considered in this and many other papers, a 2-dimensional square lattice is used. An example for a protein conformation under the 2D HP model is shown in Figure 1.

One of the most common approaches to protein structure prediction is to model the free energy of the given amino-acid chain depending on its conformation and then to find energy-minimising conformations. In the HP model, based on the biological motivation given above, the energy of a conformation is defined as the number of topological contacts between hydrophobic amino-acids that are not neighbours in the given sequence. More specifically, a conformation c with exactly n such H-H contacts has free energy $E(c) = n \cdot (-1)$; *e.g.*, the conformation shown in Figure 1 has energy -9.

The 2D HP protein folding problem can be formally defined as follows: Given an amino-acid sequence $s = s_1 s_2 \dots s_n$, find an energy-minimising conformation of s, *i.e.*, find $c^* \in C(s)$ such that $E(c^*) = \min\{E(c) \mid c \in C\}$, where $C(s)$ is the set of all valid conformations for s. It was recently proven that this problem and several variations of it are NP-hard [8].

Seq. No.	Length	E^*	Protein Sequence
1	20	**-9**	hphpphhphpphphhpphph
2	24	**-9**	hhpphpphpphpphpphpphpphh
3	25	**-8**	pphpphhpppphhpppphhpppphh
4	36	-14	ppphhpphhpppppphhhhhhhpphhpppphhpphpp
5	48	-23	pphpphhpphhpppppphhhhhhhhhhpppppphhpphhpphpphhhhh
6	50	-21	hhphphphphhhhphpppphpppphpppphpppphpppphphhhhphphphphh
7	60	-36	ppphhphhhhhhhhpppphhhhhhhhhhphpppphhhhhhhhhhhhpppphh hhhhphhph
8	64	-42	hhhhhhhhhhhhphphpphhpphhpphpphhpphhpphpphhpphhpphp hphhhhhhhhhhhh
9	20	**-10**	hhhpphphphpphphphpph

Table 1. Benchmark instances for the 2D HP protein folding problem used in this study with known or approximated optimal energy values E^*. (E^* values printed in bold-face are provably optimal.) These instances can also be found at http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html.

Existing 2D HP Protein Folding Algorithms

A number of well-known heuristic optimisation methods have been applied to the 2D HP protein folding problem, including Simulated Annealing (SA) [15] and Evolutionary Algorithms (EAs) [8, 18, 10, 17]. The latter have been shown to be particular robust and effective for finding high-quality solutions to the 2D HP protein folding problem [8].

An early application of EAs to protein structure prediction was presented by Unger and Moult [17, 18]. They presented a nonstandard EA incorporating characteristics of Simulated Annealing. Using an algorithm that searches in a space of conformations represented by *absolute* directions and considers only feasible configurations (self-avoiding paths on the lattice), Unger and Moult were able to find high-quality conformations for a set of protein sequences of length up to 64 amino-acids (see Table 1; we use the same benchmark instances for evaluating our ACO algorithm). Unfortunately, it is not clear how long their algorithm ran to achieve these results.

Krasnogor *et al.* [7] implemented another EA in which the conformations are represented using relative folding directions or local structure motifs – the same representation used by our algorithm. Their algorithm found the best known conformations for Sequences 1 through 6 and 9 from Table 1. The best value they achieved for Sequences 7 and 8 were -33 and -39, respectively.

Among the best known algorithms for the 2D HP protein folding problem are various Monte Carlo methods, including the Pruned Enriched Rosenbluth Method of Bastolla *et al.* [1]. Using this method, the best known solution of Sequence 7 ($E^* = -36$) could be found; however, even it failed to obtain the best known conformation for Sequence 8. Other state-of-the-art methods for this problem include the dynamic Monte Carlo algorithm by Ramakrishnan *et al.* [12] and the evolutionary Monte Carlo algorithm by Liang *et al.* [11]. The Core-

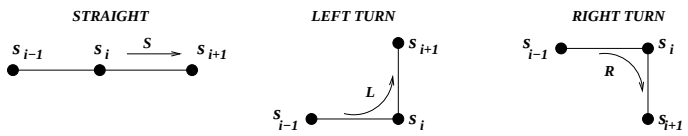

Fig. 2. The local structure motifs which form the solution components underlying the construction and local search phases of our ACO algorithm.

directed Chain Growth method by Beutler *et al.* was able to find ground states for all benchmark sequences used here, except for Sequence 7 [2]. Currently, none of these algorithm seems to dominate the others.

3 Applying ACO to the 2D HP Protein Folding Problem

The ants in our ACO algorithm construct candidate conformations for a given HP protein sequence and apply local search to achieve further improvements. As in [7], candidate conformations are represented using local structure motifs (or relative folding directions) *straight* (S), *left* (L), and *right* (R) which for each amino-acid indicate its position on the 2D lattice relative to its direct predecessors in the given sequence (see Figure 2). Since conformations are invariant *w.r.t.* rotations, the position of the first two amino-acids can be fixed without loss of generality. Hence, we represent candidate conformations for a protein sequence of length n by a sequence of local structure motifs of length $n-2$. For example, the conformation of Sequence 1 shown in Figure 1 corresponds to the motif sequence LSLLRRLRLLSLRRLLSL.

Construction Phase, Pheromone and Heuristic Values

In the construction phase of our ACO algorithm, each ant first randomly determines a starting point within the given protein sequence. This is done by choosing a sequence position between 1 and $n-1$ according to a uniform random distribution and by assigning the corresponding amino-acid (H or P) and its direct successor in the sequence arbitrarily to neighbouring positions on a 2D lattice. From this starting point, the given protein sequence is folded in both directions, adding one amino-acid symbol at a time. The relative directions in which the conformation is extended in each construction step are determined probabilistically using a heuristic function as well pheromone values (also called trail intensities); these relative directions correspond to local structure motifs between triples of consecutive sequence positions $s_{i-1}s_is_{i+1}$ that form the solution components used by our ACO algorithm; conceptually, these play the same role as the edges between cities in the classical application of ACO to the Travelling Salesperson Problem.

When extending a conformation from sequence position i to the right by placing amino-acid s_{i+1} on the lattice, our algorithm uses pheromone values

$\tau_{i,d}$ and heuristic values $\eta_{i,d}$ where $d \in \{S, L, R\}$ is a relative direction. Likewise, pheromone values $\tau'_{i,d}$ and heuristic values $\eta'_{i,d}$ are used when extending a conformation from position i to the left. In our algorithm, we use $\tau'_{i,L} = \tau_{i,R}$, $\tau'_{i,R} = \tau_{i,L}$, and $\tau'_{i,S} = \tau_{i,S}$. This reflects a fundamental symmetry underlying the folding process: Extending the fold from sequence position i to $i+1$ by placing s_{i+1} right of s_i (as seen from s_{i-1}) or extending it from position i to $i-1$ by placing s_{i-1} left of s_i (as seen from s_{i+1}) leads to the same local conformation of $s_{i-1}s_is_{i+1}$.

The heuristic values $\eta_{i,d}$ should guide the construction process towards high-quality candidate solutions, *i.e.*, towards conformations with a maximal number of H-H interactions. In our algorithm, this is achieved by defining $\eta_{i,d}$ based on $h_{i+1,d}$, the number of new H-H contacts achieved by placing s_{i+1} in direction d relative to s_i and s_{i-1} when folding forwards (backwards folding is handled analogously and will not be described in detail here). Note that if $s_{i+1} = P$, this amino-acid cannot contribute any new H-H contacts and hence $h_{i,S} = h_{i,L} = h_{i,R} = 0$. Furthermore, for $1 < i < n-1$, $h_{i,d} \leq 2$ and $h_{n-1,d} \leq 3$; the actual $h_{i,d}$ values can be easily determined by checking the seven neighbours of the possible positions of s_{i+1} on the 2D lattice (obviously, the position of s_i is occupied and hence not included in these checks). The heuristic values are then defined as $\eta_{i,d} = h_{i,d} + 1$; this ensures that $\eta_{i,d} > 0$ for all i and d which is important in order not to exclude *a priori* any placement of s_{i+1} in the construction process.

When extending a partial conformation $s_k \ldots s_i$ to s_{i+1} during the construction phase of our ACO algorithm, the relative direction d of s_{i+1} *w.r.t.* $s_{i-1}s_i$ is determined based on the heuristic and pheromone values according to the following probabilities:

$$p_{i,d} = \frac{[\tau_{i,d}]^\alpha [\eta_{i,d}]^\beta}{\sum_{e \in \{L,R,S\}} [\tau_{i,e}]^\alpha [\eta_{i,e}]^\beta} \tag{1}$$

Analogously, when extending partial conformation $s_i \ldots s_m$ to s_{i-1}, the probability of placing s_{i-1} in relative direction d w.r.t. $s_{i+1}s_i$ is defined as:

$$p'_{i,d} = \frac{[\tau'_{i,d}]^\alpha [\eta'_{i,d}]^\beta}{\sum_{e \in \{L,R,S\}} [\tau'_{i,e}]^\alpha [\eta'_{i,e}]^\beta} \tag{2}$$

From its randomly determined starting point l, each ant will first construct the partial conformation $s_l \ldots s_1$ and then the partial conformation $s_l \ldots s_n$. We also implemented variants of our algorithm in which all ants start their construction process at the same point (left end, middle, or right end of the protein sequence). Performance results for these alternative mechanisms are reported in Section 4.

Especially for longer protein sequences, infeasible conformations are frequently encountered during the construction process. This happens if an incomplete conformation cannot be extended beyond a given lattice position because all neighbouring lattice positions are already occupied by other amino-acids. Our algorithm uses two mechanisms to address this problem: Firstly, using a simple look-ahead mechanism we never allow an "internal" amino-acid s_i $(1 < i < n)$

to be placed such that all its neighbouring positions on the grid are occupied.[1] Secondly, if during a construction step all placements of s_i are ruled out by the look-ahead mechanism, we backtrack half the distance already folded and restart the construction process from the respective sequence position.[2]

Local Search

Similar to other ACO algorithms known from the literature, our new algorithm for the 2D HP protein folding problem incorporates a local search phase. After the construction phase, each ant applies a hybrid iterative improvement local search to its respective candidate conformation. We use two types of neighbourhoods for this local search process:

- the so-called "macro-mutation neighbourhood" described in Krasnogor *et al.* [8], in which neighbouring conformations differ in a variable number of up to $n-2$ consecutive local structure motifs;
- a 1-exchange "point mutation" neighbourhood, in which two conformations are neighbours if they differ by exactly one local structure motif.

Our local search algorithm alternates between these two phases. In each iteration, first a macro-mutation step is applied to the current conformation. This involves randomly changing all local structure motifs between two randomly determined sequence positions. All changes are performed in such a way that the resulting conformation is guaranteed to be feasible, *i.e.*, remains a self-avoiding walk on the 2D lattice. If the macro-mutation step results in an improvement in energy, the local search continues from the respective conformation; otherwise, the macro-mutation step has no effect. Next, a sequence of up to $n-2$ restricted 1-exchange steps are performed. This is done by visiting all sequence positions in random order; for each position, all 1-exchange neighbours that can be reached by modifying the corresponding local structure motif are considered.

Whenever any of these yields an improvement in energy, the corresponding mutation is applied to the current conformation. These local search iterations are repeated until no improvements in solution quality have been achieved for a given number *noImpr* of search steps. (Of the various hybrid local search methods we implemented and studied, the one described here seemed to work best.)

Update of the Pheromone Values

After each construction and local search phase, selected ants update the pheromone values in a standard way:

$$\tau_{i,d} \leftarrow (1-\rho)\tau_{i,d} + \Delta_{i,d,c} \tag{3}$$

[1] This is extremely cheap computationally, since it can be checked easily during the computation of the heuristic values.

[2] Various modifications of this backtracking mechanism were tested; the one presented here proved to be reasonably fast and effective.

where $0 < \rho \leq 1$ is the pheromone persistence (a parameter that determines how fast the information gathered in previous iterations is "forgotten") and $\Delta_{i,d,c}$ is the relative solution quality of the given ant's candidate conformation c, if that conformation contains a local structure motif d at sequence position i and zero otherwise. We use the relative solution quality, $E(c)/E^*$, where E^* is the known minimal energy for the given protein sequence (or an approximation based on the number of H residues in the sequence) in order to prevent premature search stagnation for sequences with large energy values.

As a further mechanism for preventing search stagnation, we use an additional "renormalisation" of the pheromone values that is conceptually similar to the method used in MAX-MIN Ant System [16]. For a given sequence position i, whenever the ratio between the maximal and minimal $\tau_{i,d}$ values, τ_i^{max} and τ_i^{min}, falls below a threshold θ, the minimal $\tau_{i,d}$ value is set to $\tau_i^{max} \cdot \theta$ while the maximal $\tau_{i,d}$ value is decreased by $\tau_i^{max} \cdot \theta$. This guarantees that the probability of selecting an arbitrary local structure motif for the corresponding sequence position does not become arbitrarily small.

We implemented various methods for selecting the ants that are allowed to update the pheromone values, including elitist strategies known from the literature. Performance results obtained for these variants are reported in Section 4.

4 Empirical Results

To assess its performance, we applied our ACO algorithm to the nine standard benchmark instances for the 2D HP protein folding problem shown in Table 1; these are the same instances used by Unger and Moult [17, 18]. Experiments were conducted by performing a variable number of runs for each problem instance; each run was terminated when no improvement in solution quality had been observed over $10,000$ cycles of our ACO algorithm. We used 10 ants for small sequences ($n \leq 25$) and 10–15 ants for larger sequences. Unless explicitly indicated otherwise, we used the following parameter settings for all experiments: $\alpha = 1$, $\beta = 2$, $\rho = 0.6$, and $\theta = 0.05$. The local search procedure was terminated if no solution improvement had been obtained within 100–300 search steps. We used an elitist pheromone update in which only the best 20% of the conformations obtained after the local search phase were used for updating the pheromone values. Additionally, the globally best conformation was used for updating the pheromone values whenever no improvement in solution quality had been seen within 20–50 cycles. Run-time was measured in terms of CPU time and all experiments were performed on PCs with 1GHz Pentium III CPUs, 256KB cache and 1GB RAM.

As can be seen from the results reported in Table 2, our ACO algorithm found optimal solutions for all but the two longest benchmark protein sequences. For Sequence 7, we achieved the same sub-optimal solution quality as Unger and Moult's evolutionary algorithm. For sequences of length 25 and below, our algorithm found optimal solutions in each of multiple attempts, while for longer protein sequences often many solution attempts were required. Following the

Instances			*ACO + Local Search*					*Local Search Only*			
Seq. No.	Length	E^*	sq	n_{opt}/n_{runs}	% *suc.*	t_{avg}	sq	n_{opt}/n_{runs}	% *suc.*	t_{avg}
1	20	**-9**	**-9**	711/711	100.0	23.90	**-9**	100/258	38.7	111.43
2	20	**-9**	**-9**	596/596	100.0	26.44	**-9**	8/113	7.0	162.15
3	25	**-8**	**-8**	120/120	100.0	35.32	**-8**	44/129	34.1	125.42
4	36	-14	**-14**	21/128	16.4	4746.12	**-14**	5/72	6.9	136.10
5	48	-23	**-23**	1/151	0.6	1920.93	-21	1/20	5.0	1780.74
6	50	-21	**-21**	18/43	41.9	3000.28	-20	3/18	16.7	1855.96
7	60	-36	-34	1/119	0.8	4898.77	-33	2/20	10.0	1623.21
8	64	-42	-32	1/22	4.5	4736.98	-33	2/9	22.2	1441.88
9	24	**-10**	**-10**	247/247	100.0	43.48	**-10**	5/202	25.0	134.57

Table 2. Comparison of the local search and the ACO, where sq is the best solution quality over all runs, n_{opt} is the number of runs the algorithm finds sq, n_{runs} is the total number of runs, % *suc.* is the percentage of runs in which solution quality sq was achieved, and t_{avg} is the average CPU time [sec] required by the algorithm to find sq.

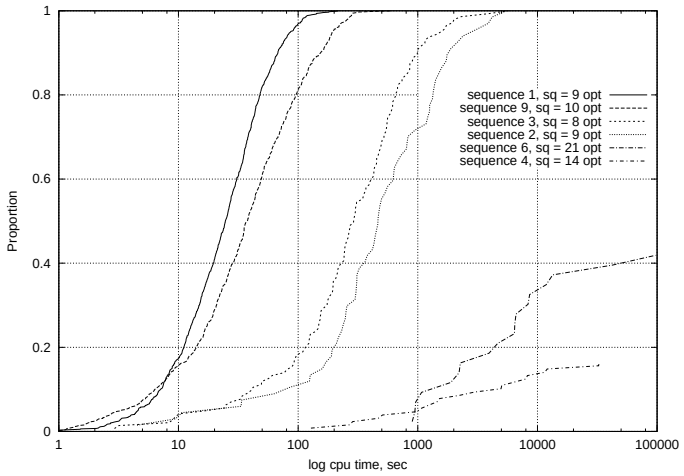

Fig. 3. Run-time distributions of our ACO algorithm applied to several benchmark instances; note stagnation behaviour for large instances.

methodology of Hoos and Stützle [6], we measured run-time distributions (RTD) of our ACO algorithm; for all sequences in which our algorithm found the best known conformation more than once, the respective RTDs are shown in Figure 3. Evidence of search stagnation behavior can be clearly observed for large sequences; in these cases, using a better construction heuristic and/or more aggressive local search may help to improve performance.

To better understand the role of ACO as compared to the local search method in the optimisation process, we also performed experiments in which only the local search method was applied to the same benchmark instances. As seen

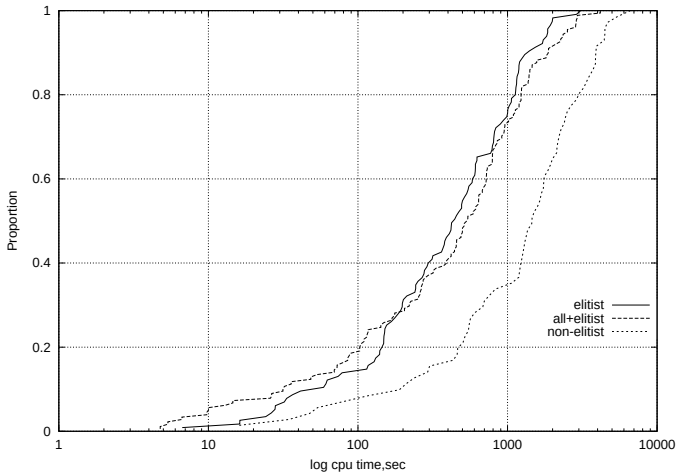

Fig. 4. RTDs for ACO with elitist, all+elitist, and non-elitist pheromone update applied to benchmark instance 2.

in Table 2, local search typically takes longer and in many cases fails to find solutions of the same quality as our ACO algorithm. In experiments not reported here, we also found that ACO without a local search takes substantially more time (cycles) to reach high-quality solutions than ACO with local search.

In order to evaluate the benefit from using a population of ants instead of just a single ant, we studied the impact of varying the number of ants on the performance of our algorithm. While using a single ant only, we still obtained good performance for small problems ($n \leq 25$), the best known solution to Problem 4 ($n = 36$) could not been found within 2 CPU hours.

It has been shown for other applications of ACO that elitist pheromone update strategies can lead to performance improvements. The same appears to be the case here: We tested three different pheromone update strategies: non-elitist – all ants update pheromone values; all+elitist – same, but the best 20% conformations are additionally reinforced; and elitist only – only the best 20% of the conformations are used for updating the pheromone values in each cycle. As can be seen in Figure 4, elitist update results in considerably better performance than non-elitist update. For larger sequences ($n \geq 36$), the best known solution qualities could not been obtained within 2 CPU hours when using non-elitist update. This suggests that the search intensification provided by elitist pheromone update is required for achieving good performance of our ACO algorithm. At the same time, additional experiments (not reported here) indicate that our pheromone renormalisation mechanism is crucial for solving large problem instances, which underlines the importance of search diversification.

In the next experiment, we investigated the influence of pheromone values compared to heuristic information on performance. As illustrated in Figure 5,

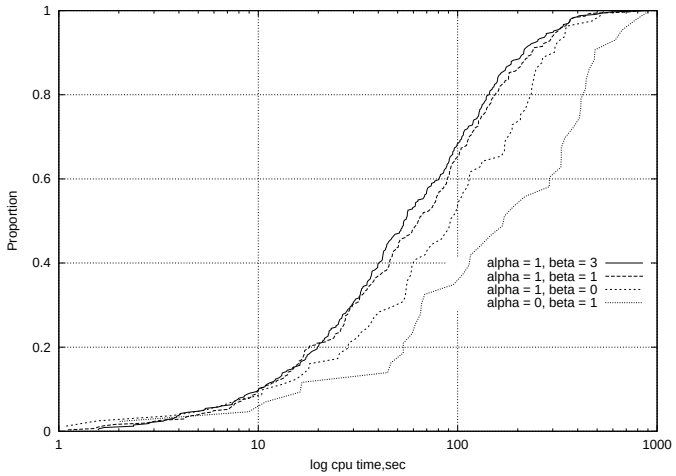

Fig. 5. RTDs for our ACO applied to Sequence 1, using various values of α and β.

the results show that both, pheromone values and heuristic information are important for achieving good performance. Both extreme cases, $\alpha = 0$, *i.e.*, pheromone values are ignored, and $\beta = 0$, *i.e.*, heuristic values are ignored, lead to performance decreases even for small problem instances. Interestingly, using pheromone information only is less detrimental than solely using heuristic information. This phenomenon becomes more pronounced on larger problem instances; *e.g.*, when ignoring pheromone information ($\alpha = 0$), our ACO algorithm was not able to find the best known solution to Sequence 4 within 3 CPU hours.

Finally, we studied the effect of the starting point for the construction of conformations on the performance of our ACO. It has been shown that real proteins fold by hierarchical condensation starting from folding nuclei; the use of complex and diverse folding pathways helps to avoid the need to extensively search large regions of the conformation space [14]. This suggests that the starting point for the folding process can be an important factor in searching for optimal conformations. We tested four strategies for determining the starting point for the folding process performed in the construction phase of our algorithm: all ants fold forwards, starting at sequence position 1; all ants fold backwards, starting at sequence position n; all ants fold forwards and backwards, starting in the middle of the given sequence; and all ants fold forwards and backwards, starting at randomly determined sequence positions (in which case all ants can fold from different starting points). As can be seen from Figure 6, the best performance is obtained by letting all ants start the folding process from individually selected, random sequence positions. This result is even more prominent for longer sequences and suggests that the added search diversification afforded by multiple and diverse starting points is important for achieving good performance.

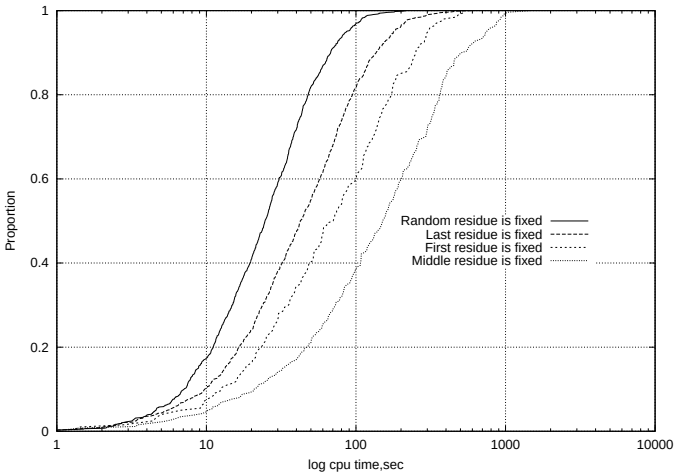

Fig. 6. RTDs for our ACO for Sequence 1, using various strategies for choosing the starting point for constructing candidate conformations.

5 Conclusions and Future Work

In this paper we introduced an ACO algorithm for the 2D HP protein folding problem, an extremely simplified but widely studied and computationally hard protein structure prediction problem, to which to our best knowledge, ACO has not been previously applied. An empirical study of our algorithm demonstrated the effectiveness of the ACO approach for solving this problem and highlighted the impact of various features of our algorithm, including elitist pheromone update and randomly chosen starting points for the folding process.

In this study we presented first evidence that ACO algorithms can be successfully applied to protein folding problems. There are many directions for future research. Clearly, there is substantial room for improvement in the local search procedure. In preliminary experiments with a conceptually simpler local search procedure designed to minimise the occurrence of infeasible configurations we have already observed significant improvements over the results presented here. Furthermore, different heuristic functions should be considered; in this context, techniques that allow the approximation of the size of a protein's hydrophobic core are promising. It might also be fruitful to consider ACO approaches based on more complex solution components than the simple local structure motifs used here. Finally, we intend to develop and study ACO algorithms for other types of protein folding problems, such as the 3-dimensional HP model in the near future [17]. Overall, we strongly believe that ACO algorithms offer considerable potential for solving protein structure prediction problems robustly and efficiently and that further work in this area should be undertaken.

References

1. Bastolla U., H. Fravenkron, E. Gestner, P. Grassberger, and W. Nadler. *Testing New Monte Carlo algorithm for the protein folding problem.* Proteins-Structure Function and Genetics 32 (1): 52-66, 1998.
2. Beutler T., and K. Dill. *A fast conformational search strategy for finding low energy structures of model proteins.* Protein Science (5), pp. 2037–2043, 1996.
3. Dorigo, M., V. Maniezzo, and A. Colorni. *Positive feedback as a search strategy.* Technical Report 91–016, Dip. Elettronica, Politecnico di Milano, Italy, 1991.
4. Dorigo, M. and G. Di Caro. The ant colony optimization meta-heuristic. In *New Ideas in Optimization*, pp. 11–32. McGraw-Hill, 1999.
5. Dorigo, M., G. Di Caro and L.M. Gambardella. *Ant Algorithms for Discrete Optimization.* Artificial Life, 5,2, pp. 137–172, 1999.
6. Hoos, H.H., and T. Stützle. *On the empirical evaluation of Las Vegas algorithms.* Proc. of UAI-98, Morgan Kaufmann Publishers, 1998. IEICE Trans. Fundamentals, Vol. E83-A:2 ,Feb. 2000.
7. Krasnogor, N., D. Pelta, P. M. Lopez, P. Mocciola, and E. de la Canal. *Genetic algorithms for the protein folding problem: a critical view.* In C.F.E. Alpaydin, ed., Proc. Engineering of Intelligent Systems. ICSC Academic Press, 1998.
8. Krasnogor, N., W.E. Hart. J. Smith, and D.A. Pelta. *Protein structure prediction with evolutionary algorithms.* Proceedings of the genetic & evolutionary computation conference, 1999.
9. Lau, K.F., and K.A. Dill. *A lattice statistical mechanics model of the conformation and sequence space of proteins.* Macromolecules 22:3986–3997, 1989.
10. Patton, A.W.P. III, and E. Goldman. *A standard GA approach to native protein conformation prediction.* In Proc. 6^{th} Intl. Conf Genetic Algorithms, pp. 574–581. Morgan Kauffman, 1995.
11. Liang F., and W.H. Wong. *Evolutionary Monte Carlo for protein folding simulations.* J. Chem. Phys. 115 (7), pp. 3374–3380, 2001.
12. Ramakrishnan R., B. Ramachandran, and J.F. Pekny. *A dynamic Monte Carlo algorithm for exploration of dense conformational spaces in heteropolymers.* J. Chem. Phys. 106 (6), 8 February, pp. 2418–2424,1997.
13. Richards, F. M. *Areas, volumes, packing, and protein structures.* Annu. Rev. Biophys. Bioeng. 6:151–176, 1977.
14. Rose, G. D. *Hierarchic organization of domains in globular proteins.* J. Mol. Biol. 134:447–470, 1979.
15. Sali, A., E. Shakhnovich and M. Karplus, *How Does a Protein Fold?* Nature, 369, pp. 248–251, May 1994.
16. Stützle, T., and H.H. Hoos. *Improvements on the ant system: Introducing MAX-MIN ant system.* In Proc. Intel. Conf. on Artificial Neural Networks and Genetic Algorithms, pp. 245–249. Springer Verlag, 1997.
17. Unger, R., and J. Moult. *A genetic algorithm for three dimensional protein folding simulations.* In Proc. 5^{th} Intl. Conf. on Genetic Algorithms, pp. 581–588. Morgan Kaufmann, 1993.
18. Unger, R., and J. Moult. *Genetic algorithms for protein folding simulations.* J. of Molecular Biology 231 (1): 75–81, 1993.