Enhanced Population Based Ant Colony for the 3D Hydrophobic Polar Protein Structure Prediction Problem

Sara Salah¹, Abdel-Rahman Hedar² and Marghny. H. Mohammed³

¹ Computer Science Department, Faculty of Science, Assiut University
² Computer Science Department, Faculty of Computers and Information, Assiut University
³ Computer Science Department, Faculty of Computers and Information, Assiut University

Abstract

Population-based Ant Colony algorithm is stochastic local search algorithm that mimics the behavior of real ants, simulating pheromone trails to search for solutions to combinatorial optimization problems. This paper introduces population-based Ant Colony algorithm to solve 3D Hydrophobic Polar Protein structure Prediction Problem then introduces a new enhanced approach of population-based Ant Colony algorithm called Enhanced Population-based Ant Colony algorithm (EP-ACO) to avoid stagnation problem in population-based Ant Colony algorithm and increase exploration in the search space escaping from local optima. The experiments show that our approach appears more efficient results than state of art method.

Keywords

Population based ACO, Ant Colony Optimization, HP Model, Protein Structure Prediction

1. Introduction

Recent breakthroughs in DNA and protein sequencing have unlocked many secrets of molecular biology. A complete understanding of gene function, however, requires a protein structure in addition to its sequence. Accordingly, the better we understand how proteins are built, the better we can deal with many common diseases. In particular, information on structural properties of proteins can give insight into the way they work and therefore help and influence modern medicine and drug development [1].

The protein structure prediction problem (PSP) is that of computationally predicting the three dimensional structure of protein from the sequence of amino acids alone. This has been an open problem for more than 30 years and developing a practical solution is widely considered the 'holy grail' of computational biology. The various approaches to the problem can be classified into two categories: knowledge based methods building the structure based on knowledge of a good template structure [16]; Ab initio methods building the structure from scratch using primary principles. Ab initio do not rely on known structures in the PDB [2] as...
knowledge based methods, instead, they predict the 3D structure of proteins given their primary sequences only. The underlying strategy is to find the best stable structure based on a chosen energy function. According to Anfinsen famous hypothesis, a protein native structure is determined by its sequence which corresponds to minimum Gibbs energy [3]. The main challenge of this approach is to search for the most stable structure in a huge search space. In general, Ab initio PSP can be reduced to the following three steps:

1) Design a simple model with a desired level of accuracy to represent the protein structure, when we approach PSP problem, probably the first thing is to represent protein structure in the problem space we can represent protein structures using two categories: All-atom Model and Simplified Models.
   - All-atom model: protein structures are represented by lists of 3D coordinates of all atoms in a protein. Although an accurate all-atom model is desired in the structure prediction, it causes too huge a computation overhead even for very small proteins.
   - Simplified Models: simplified models can be classified into lattice models and off-lattice models. Lattice models adopted lattice environment which is a grid and structural elements are positioned only at grid intersections; whereas off-lattice models use off-lattice environment in which structural elements are positioned in a continuous space.

In this paper we concern on lattice models, perhaps the simplest lattice protein model is Hydrophobic Polar (HP Model). It was proposed by Dill [4] and is widely studied for Ab initio prediction.

2) Define an energy function that can effectively discriminate native states from nonnative states. The HP model is based on the observation that the Hydrophobic Force is the main force for protein folding more about energy function we will discuss in section 2.

3) Design an efficient algorithm to find minimal energy conformations easily.

Ant Colony Optimization (ACO) [5] [6], a non-deterministic algorithm, aims to mimic the behaviors of real ant colonies to solve real-world optimization problems. ACO algorithms are a class of constructive heuristic algorithms, which build solutions to a given optimization problem, one solution component at a time, according to a defined set of rules (heuristics), starting with an empty solution add solution components until a complete solution is built. One of the main characteristics of an ACO algorithm is the pheromone information which stores information on good solutions that have been found by ants of former iterations. The pheromone information is what is transferred from one iteration of the algorithm to the next. An alternative scheme was introduced called Population based ACO (P-ACO) instead of pheromone information as in ACO, in P-ACO a population of solutions is transferred from one iteration of the algorithm to the next.

In this paper we provide a description of P-ACO algorithm and applying it for the first time to solve 3D HP lattice protein structure prediction problem then introduce new approach of population based ant colony algorithm called Enhanced Population Ant Colony (EP-ACO) to avoid stagnation problem in P-ACO algorithm. The experimental results based on different test cases of the PSP show that our algorithm enhances the performance of P-ACO.
The paper is organized as follows: Section 2 describes the HP model and mentions some heuristics algorithms form the literature for the protein structure prediction problem in the 3D HP model. An introduction to Population based ACO is given in Section 3. Our algorithm EP-ACO and application in the 3D HP Protein structure prediction is described in Section 4. The experiments and the results are presented in section 5. Conclusions are given in Section 6.

2. The HP model

The HP model is considered as the simplest abstraction of the PSP problem. This model divides the 20 standard amino acids into only two classes, according to their affinity to water: Hydrophobic amino acid is represented by (H) and Polar amino acid is represented by (P) as shown in Table 1.

Table 1. The used Hydrophobic-Polar classification of amino acids takes from [7].

| Hydrophobic (H) | Hydrophilic/Polar (P) |
|----------------|----------------------|
| C : Cysteine   | A : Alanine          |
| F : Phenylalanine | D : Aspartic Acid |
| I : Isoleucine | E : Glutamic Acid    |
| L : Leucine    | G : Glycine          |
| M : Methionine | H : Histidine        |
| V : Valine     | K : Lysine           |
| W : Trytophan  | N : Asparagine       |
| Y : Tyrosine   | P : Proline          |
| Q : Glutamine  | R : Arginine         |
| S : Serine     | T : Threonine        |

The folding of amino acid sequences is represented in a lattice, usually used in either square lattice (for the bi-dimensional model- 2D HP) or cubic lattice (for the three-dimensional model-3D HP). Thus, each amino acid occupies one lattice site, connected to its chain neighbors. After each amino acid takes one site on lattice, then it will form a shape that is considered to be the conformation (structure), this confirmation must be self avoiding walk to be valid. An example for a protein conformation under the 3D HP model is shown in Figure 1. There are several common ways to represent protein sequence \( s \in \{H,P\} \) on lattice, like Cartesian Coordinates, Internal Coordinate and Distance Geometry [8]. We concern on Internal Coordinate where a conformation is represented as a string of moving steps on the lattice from one amino acid to next one. There are two types of Internal Coordinate relative encoding and absolute encoding, in our study we use absolute encoding where the protein sequence is encoded as a string of character of absolute direction. The coordination number of the 3D lattice model is six, (each point has six neighbors). Thus there are six possible absolute moves from a given location. When we use absolute encoding the candidate solutions are represented as a string of characters \( \{R,L,B,F,U,D\}^{n-1} \) representing the six directions: Right, Left, Backward, Forward, Up and Down, where \( n \) is the length of the protein sequence. The example in Figure 1(a) shows a confirmation of protein sequence S1 in Table 2. Its string representation would be BUBFRDLDURULULDFD.

In the HP model, the energy of a conformation is defined as a number of topological contacts between H amino acids that are neighbors in the conformation but not successive on the protein
sequence, more specifically, a confirmation $c$ that has number $n$ of H-H contacts has free energy $E(c) = n(-1)$ as shown in Figure 1(b).

![Figure 1. (a) A sample protein conformation on the 3D HP model which corresponds to protein sequence $S_1$ in Table 2, green circle represents H amino acids while white circle symbolizes P amino acids, (b)The energy of this conformation is the number of H-H contacts that are neighbors on conformation and not successive on sequence, indicated in the Figure by dashed lines so the energy will be -11](image)

The PSP problem can be formally defined as follows given an amino acid sequence $s = \{s_1, s_2, \ldots, s_n\}$, where each amino acid in sequence $s$ is one of two classes H or P, find an energy minimizing conformation of $s$, i.e. find $c^* \in C(s)$ such that $E^* = E(c^*) = \min \{E(c) | c \in C\}$, where $C(s)$ is a set of all valid conformations for $s$. It was recently proved that this problem and several variations of it are NP-hard combinatorial optimization problem [9] [10]. A number of well known heuristic optimization methods have been applied to solve PSP in 3D HP lattice model. These include: Cutello and Nicosia [8] introduce an Immune Algorithm (IA) based on the clonal selection principle; they employ a new aging operator and specific mutation operators. Shmygelska and Hoos [11] use Ant Colony Optimization (ACO) with Local Search which consists of long-range mutation moves to improve diversity on the solutions. Lin [12] introduces a hybrid of Genetic Algorithm and Particle Swarm Optimization in order to solve PSP on 3D HP lattice. Lin [15] presents a modified artificial bee colony algorithm for protein structure prediction on lattice models.

3. Population Based ACO (P-ACO)

P-ACO has been proposed by Guntsch and Middendorf [13] and it is introduce a new way for updating pheromone matrix, (where as genetic algorithm) a population of solutions is directly transferred to next iteration, these solutions are then used to compute pheromone information for ants of new iteration where for every solution in the population some amount of pheromone added to corresponding edges. In more detail the first generation of ants works in the same way as in standard ant colony algorithm i.e. the ants search solutions using the initial pheromone matrix. But no pheromone evaporation is done. The best solution is then put in the (initially empty) population $Q$. After $k$ generations there exactly $k$ solutions in the population. From generation $k+1$ on:
One solution $Q_{out}$ must leave population, the solution leaves the population is decided by update strategy and when ever solution $Q_{out}$ leave population the corresponding amount of pheromone is subtracted from the elements of the pheromone matrix which called negative update (i.e. it correspond to evaporation).

$$\tau_{rs} = \tau_{rs} - \Delta \quad \forall a_{rs} \in Q_{out}. \quad (1)$$

One solution $Q_{in}$ is entering population and some amounts of pheromone are added to the edges presented that solution which called positive update.

$$\tau_{rs} = \tau_{rs} + \Delta \quad \forall a_{rs} \in Q_{in}. \quad (2)$$

The amount $\Delta$ which added or subtracted from pheromone matrix is defined as numerical number.

The algorithm representation of P-ACO is provided in Figure 2.

```
Algorithm P-ACO
1: Uniformly initialize pheromone map values
2: Construct m solutions
3: Select best solution and insert into population
4: Update pheromone values using Equation (2)
5: while stopping criterion not met do
6: Construct m solutions
7: Select best solution
8: if Best Solution is better than old Solution then
9: Remove old solution pheromone map using Equation (1)
10: Replace old solution
11: Add new solution pheromone map using Equation (2)
12: else
13: if currentPopulationSize < k - 1 then
14: Insert best solution into population
15: Insert best solution pheromone
16: else
17: Remove oldest solution pheromone map
18: Remove oldest solution from population
19: Add best solution to population
20: Add best solution pheromone map
21: end if
22: end if
23: end while
```

Figure 2. P-ACO algorithm

A subtle difference between ACO and P-ACO is the introduction of the solution storage and the pheromone update process. There are many update strategies to decide which solution will be deleted from the population and which solution will be remain in P-ACO algorithm, in quality update strategy if a new solution is better (in terms of quality) than the worst quality member of the population, the new solution replaces the worst solution, otherwise there is no change to the population. The aim of this strategy is that the population will retain good solutions which may have been found earlier in the search process. A possible weakness of
this strategy is that there is no way to ensure that the population does not end up with what are essentially multiple copies of the same solution.

4. **Our Algorithm Enhanced Population Ant Colony (EP-ACO)**

P-ACO is developed especially for dynamic problem such as DTSPs, but the stagnation behavior remains unsolved since identical ants may be stored in the population memory and generate high intensity of pheromone to a single trail. In our algorithm we try to avoid early stagnation by maintain a certain level of diversity in the population by adding two main aspects:

1) P-ACO has a strong exploitation capability that allows a fast convergence to a good quality solution. However, its exploration during the search may be insufficient. We add procedure to enhance the exploration of new area of search space called Segmentation where we select best solution in the population and cut it into segments and refold random segments of this solution trying to find new solution on new area of search space.

2) As in Max-Min Ant System [14] to avoid stagnation, we restart the P-ACO by re-initializing the pheromone values after $r$ iterations without improvement.

The basic idea of our algorithm, as follow: after creating the initial population, the main loop repeated until termination condition reached, where $m$ solution is constructed, if the cost of the best of $m$ solution less than the cost of the worst in the population the P-ACO update rule is used, then Segmentation procedure is begin by selecting best individual of the population and repeat for $s$ times; select random point and cut the solution from this random point to segments, select one segment randomly and refold this segment, finally, if the cost of the new solution is less than cost of the worst solution in the population the P-ACO update rule is used. The main EP-ACO algorithm is shown in Figure 3

```
Algorithm EP-ACO
1: Uniformly initialize pheromone map values
2: Construct $m$ solutions
3: Select best solution and insert into population
4: Update pheromone values using Equation (2)
5: while stopping criterion not met do
6: Construct $m$ solutions
7: Select best solution
8: if best solution is better than worst solution then
9: Remove old solution pheromone map using Equation (1)
10: Remove old solution
11: Add new solution pheromone map using Equation (2)
12: Add best solution to the population
13: end if
14: construct new solution by call Segmentation procedure
15: if new solution is better than worst solution
16: use P-ACO update rule Equation (1) and Equation (2)
17: end if
18: Reinitialize pheromone map every $r$ iteration
19: end if
20: end while
```
When applying P-ACO to 3D HP model; we want to know how to construct solution and then follow the storage of solutions and update pheromone rules in Figure 2, in construct solution step: ants start to construct solution. There are six possible positions on the 3D lattice for every amino acid. They are the neighbor positions of the precedence amino acid. Since conformations are rotationally invariant, the position of the first two amino acids can be fixed without loss of generality. During the construction phase, ants fold a protein from the left end of the sequence adding one amino acid at a time based on the two sources of information: pheromone matrix value, and heuristic information. The transition probability to select the position of the next amino acid is given as:

\[
p_{i,d} = \frac{\eta_i^a \cdot \tau_i^b}{\sum_{e \in \{U,D,L,R,F,B\}} \eta_e^a \cdot \tau_e^b},
\]  

(3)

where \( \alpha \) and \( \beta \) are parameters that determine the relative influence of pheromone and heuristic information. The pheromone values \( \tau_{i,d} \) indicate the amount of pheromone deposited by each ant on the path \((i,d)\), the heuristic function \( \eta_{i,d} \) used here as illustrated in section 2.

5. Experiments and Results

In this section we apply P-ACO and EP-ACO to solve PSP on 3D HP lattice model, first a comparative study between simple P-ACO approach and EP-ACO is done to show the performance of our algorithm. Then the behavior of EP-ACO is compared with state of art methods used to solve this problem.

For the following experiment results, All experiments were performed on PCs with 2 GHz Intel core(TM)2 due CPU and 2 MB RAM, running windows 7 (our reference machine), the program was written using java program language and run-time was measured in terms of CPU time. Table 2 presents 3D HP instances considered for the computational experiments taken from [8]. For each HP sequence, the column Instance represents the sequence number; the Length represents the number of amino acid in the protein sequence.
Table 2. HP instances

| Instance | Length | Protein Sequence |
|----------|--------|------------------|
| S1       | 20     | HPHPHHPPHPHPPHPH |
| S2       | 24     | HHPPHPHPHPHPHPH |
| S3       | 25     | PPHPHPHPHPPPPHPPH |
| S4       | 36     | PPPHPHPHPHPPPPHPP |
| S5       | 48     | PPHPHPHPPPPPPPPPPP |  
| S6       | 50     | HHPHPHPPHPHPPPPPH |
| S7       | 60     | PPHPHPPPPPPPPPPP |  
| S8       | 64     | HHHHHHHHHHHHHHHP |

Table 3, presents comparison between P-ACO and our proposed EP-ACO. The parameters settings for the P-ACO and our proposed EP-ACO are $\alpha = 1$, $\beta = 3$, $\tau_0 = 1$ number of ants $m = 100$, pop size =10 and $\Delta$ is equal to the energy of that solution that will be added or removed, for EP-ACO $s = 50$ for small sequence length ($n < 48$) for larger sequence length ($n > 48$) we set $s = 100$, finally, we reinitialize the pheromone after 3000 iteration. Each run of algorithm ends when the maximum number of evaluation to fitness function is equal to $10^6$.

All the experimental results reported in Table 3 are averaged over 30 independent runs. The column Best means the best found energy (Fitness) value; the Mean is the mean of energy found over 30 independent runs. As shown in Table 3, our algorithm EP-ACO achieves best result than P-ACO.

Table 3. Comparison between P-ACO algorithm and EP-ACO algorithm

| Instance | P-ACO | EP-ACO |
|----------|-------|--------|
|          | Best  | Mean   | Best  | Mean  |
| S1       | -11   | -11    | -11   | -11   |
| S2       | -13   | -12.6  | -13   | -13   |
| S3       | -9    | -9     | -9    | -9    |
| S4       | -18   | -17.8  | -18   | -18   |
The performance of the proposed model is compared to the best results obtained by other algorithms for protein structure prediction in 3D HP model. Table 4 presents the results as follows: the best energy found by the proposed method (in the last column), the results of Protein 3D HP Model Folding Simulation Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization (HGAPSO) [12], Immune Algorithm for Protein Structure Prediction (IA) [8] and Artificial Bee Colony Algorithm For Protein Structure Prediction On Lattice Models (MABC) [15]. As shown in Table 4, the EP-ACO model is able to identify the protein configurations having the best Fitness Energy for sequences S5 to S8. The structure of 8 protein sequences can be clearly seen in Figure 4.

Table 4. The simulation results obtained from the proposed algorithm compared with the methods given in the literature. Figures in bold indicate the lowest energy

| Instance | HGA-PSO | IA | MABC | EP-ACO |
|----------|---------|----|------|--------|
|          | Best    | Best | Best | Best   |
| S1       | -11     | -11 | -11  | -11    |
| S2       | -13     | -13 | -13  | -13    |
| S3       | -9      | -9  | -9   | -9     |
| S4       | -18     | -18 | -18  | -18    |
| S5       | -29     | -29 | -29  | -31    |
| S6       | -26     | -23 | -26  | -32    |
| S7       | -49     | -41 | -49  | -54    |
| S8       | -       | -42 | -    | -58    |
Figure 4. Results of the structure of 8 protein sequence
6. Conclusion

In this paper, we presented the Population Based Ant Colony algorithm for 3D HP Protein Structure Prediction Problem then introduce a new approach called Enhanced Population-based Ant Colony algorithm (EP-ACO) to avoid stagnation problem in population-based Ant Colony algorithm and increase exploration in the search space escaping from local optima. It shown experimentally that our algorithm EP-ACO achieves on nearly all test sequences comparable results to other state of the art algorithms and is much better than simple P-ACO algorithm.

References

[1] Qatawneh, S., Alneaimi, A., Rawashdeh, T., Muhairat, M., Qahwaji, R., and Ipson, S. (2012) ‘Efficient Prediction of DNA-Binding Proteins Using Machine Learning’, International Journal on Bioinformatics & Biosciences, arXiv preprint arXiv:1207.2600.
[2] Berman, H., Henrick, K. and Nakamura, H. (2003) ‘Announcing the worldwide protein data bank’, Nature Structural & Molecular Biology, 10(12), 980-980.
[3] Anfinsen, C. B. (1973) ‘Principles that govern the folding of protein chains’, Science, 181(96), 223-230.
[4] Dill, K. A., Bromberg, S., Yue, K., Fiebig, K. M., Yee, D. P., Thomas, P. D. and Chan, H. S. (1995) ‘Principles of protein folding--a perspective from simple exact models’, Protein science: a publication of the Protein Society, 4(4), 561.
[5] Dorigo, M., Caro, G. D. and Gambardella, L. M. (1999) ‘Ant algorithms for discrete optimization’, Artificial life, 5(2), 137-172.
[6] Dorigo, M. and Gambardella, L. M. (1997) ‘Ant colony system: A cooperative learning approach to the traveling.
[7] Ullah, A. D., Kapokalivas, L., Mann, M. and Steinhöfel, K. (2009) ‘Protein folding simulation by two-stage optimization’, In Computational Intelligence and Intelligent Systems. Springer BerliHeidelberg, 138-145.
[8] Cutello, V., Nicosia, G., Pavone, M. and Timmis, J. (2007) ‘An immune algorithm for protein structure prediction on lattice models’, Evolutionary Computation, IEEE Transactions on, 11(1), 101-117.
[9] Unger, R. and Moult, J. (1993) ‘Finding the lowest free energy conformation of a protein is an NP-hard problem’, proof and implications. Bulletin of Mathematical Biology, 55(6), 1183-1198.
[10] Moult, J. (1993) ‘A genetic algorithm for 3D protein folding simulations’, In Genetic Algorithms, Morgan Kaufmann Publishers, 581.
[11] Shmygelska, A. and Hoos, H. H. (2005) ‘An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem’, BMC bioinformatics, 6(1), 30.
[12] Lin, C. J. and Su, S. C. (2011) ‘Protein 3D HP Model Folding Simulation Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization’, International Journal of Fuzzy Systems, 13(2), 140-147.
[13] Guntsch, M. and Middendorf, M. (2002) ‘A population based approach for ACO’, In Applications of Evolutionary Computing. Springer Berlin Heidelberg., 72-81.
[14] Stützle, T. and Hoos, H. H. (2000) ‘MAX–MIN ant system’. Future generation computer systems, 16(8), 889-914.
[15] Lin, C. J., and Su, S. C. (2012) ‘Using an efficient artificial bee colony algorithm for protein structure prediction on lattice models’, International journal of innovative computing, information and control, 8, 2049-2064.
[16] Jehangir, M. and Ahmad, S. F. (2013) ‘Structural studies of aspartic endopeptidase pep2 from neosartorya fisherica using homology modeling techniques’, International Journal on Bioinformatics & Biosciences.
Abdel-Rahman Hedar, received his Ph.D. degree in computer science from the University Kyoto, Japan, in 2004, his MSc and BSc From Assiut university, Assiut, Egypt, in 1997 and 1993, respectively. He is Director of Quality Assurance Unit, Associate Professor, Computer Science Department, Faculty of Computers and Information, Assiut University.

Marghny H. Mohamed, received his Ph.D. degree in computer science from the University of Kyushu, Japan, in 2001, his MSc and BSc From Assiut university, Assiut, Egypt, in 1993 and 1988, respectively. He is currently an associate professor in the Department of Computer Science, and Vice Dean for Education and Student Affairs of the Faculty of Computers and Information Systems, University of Assiut, Egypt.