STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVI**, Number 1, 2011

AN EXPERIMENT ON PROTEIN STRUCTURE PREDICTION USING REINFORCEMENT LEARNING

GABRIELA CZIBULA, MARIA-IULIANA BOCICOR AND ISTVAN-GERGELY CZIBULA

Abstract. We are focusing in this paper on investigating a reinforcement learning based model for solving the problem of predicting the bidimensional structure of proteins in the hydrophobic-polar model, a well-known *NP*-hard optimization problem, important within many fields including bioinformatics, biochemistry, molecular biology and medicine. Our model is based on a *Q-learning* agent-based approach. The experimental evaluation confirms a good performance of the proposed model and indicates the potential of our proposal.

1. Introduction

Combinatorial optimization is the seeking for one or more optimal solutions in a well defined discrete problem space. In real life approaches, this means that people are interested in finding efficient allocations of limited resources for achieving desired goals, when all the variables have integer values. As workers, planes or boats are indivisible (like many other resources), the Combinatorial Optimization Problems (COPs) receive today an intense attention from the scientific community.

The current real-life COPs are difficult in many ways: the solution space is huge, the parameters are linked, the decomposability is not obvious, the restrictions are hard to test, the local optimal solutions are many and hard to locate, and the uncertainty and the dynamicity of the environment must be taken into account. All these characteristics, and others more, constantly make the algorithm design and implementation challenging tasks. The quest

Received by the editors: March 20, 2011.

2000 *Mathematics Subject Classification.* 65K10, 68T05.

1998 *CR Categories and Descriptors.* I.2.6[**Computing Methodologies**]: Artificial Intelligence – *Learning*; I.2.8[**Computing Methodologies**]: Problem Solving, Control Methods, and Search – *Heuristic methods* .

Key words and phrases. Combinatorial optimization, Bioinformatics, Reinforcement Learning, Protein Folding.

for more and more efficient solving methods is permanently driven by the growing complexity of our world.

Yet, for COPs that are *NP*-hard, no polynomial time algorithm exists. Therefore, complete methods might need exponential computation time in the worst-case. This often leads to computation times too high for practical purposes. Thus, the use of approximate methods to solve COPs has received more and more attention. In approximate methods we sacrifice the guarantee of finding optimal solutions for the sake of getting good solutions in a significantly reduced amount of time.

Reinforcement Learning (RL) [1] is an approach to machine intelligence in which an agent can learn to behave in a certain way by receiving punishments or rewards on its chosen actions.

In this paper we aim at investigating a reinforcement learning based model for solving a well known optimization problem within bioinformatics, the problem that refers to predicting the structure of a protein from its amino acid sequence. Protein structure prediction is an *NP*-complete problem, being one of the most important goals pursued by bioinformatics and theoretical chemistry; it is highly important in medicine (for example, in drug design) and biotechnology (for example, in the design of novel enzymes).

The model proposed in this paper for solving the bidimensional *protein folding* problem can be easily extended to the problem of predicting the three-dimensional structure of proteins. Moreover, the proposed model can be generalized to address other optimization problems. To our knowledge, except for the ant based approaches [2], the bidimensional *protein structure prediction* problem has not been addressed in the literature using reinforcement learning, so far.

The rest of the paper is organized as follows. Section 2 presents the main aspects related to the *protein structure prediction* problem. The reinforcement learning model that we propose for solving the bidimensional protein folding problem is introduced in Section 3. An experiment is given in Section 4 and in Section 5 we provide an analysis of the proposed reinforcement model, emphasizing its advantages and drawbacks. Section 6 contains some conclusions of the paper and future development of our work.

2. Protein Structure Prediction. The Hydrophobic-Polar Model

The determination of the three-dimensional structure of a protein, using the linear sequence of amino acids is one of the greatest challenges of bioinformatics, being an important research direction due to its numerous applications in medicine (drug design, disease prediction) and genetic engineering (cell modelling, modification and improvement of the functions of certain proteins). Moreover, unlike the structure of other biological macromolecules (e.g., DNA), proteins have complex structures that are difficult to predict. Protein structure prediction is an important problem within the more general *protein folding* problem, and is also reffered in the literature as the *computational protein folding* problem [3]. Different computational intelligence approaches for solving the protein structure prediction problem have been proposed in the literature, so far.

An important class of abstract models for proteins are lattice-based models - composed of a lattice that describes the possible positions of amino acids in space and an energy function of the protein, that depends on these positions. The goal is to find the global minimum of this energy function, as it is assumed that a protein in its native state has a minimum free energy and the process of folding is the minimization of this energy [4].

One of the most popular lattice-models is Dill's Hydrophobic-Polar (HP) model [5].

In the folding process the most important difference between the amino acids is their hydrophobicity, that is how much they are repelled from water. By this criterion the amino acids can be classified in two categories: *hydrophobic* or *non-polar* (H) - the amino acids belonging to this class are repelled by water; *hydrophilic* or *polar* (P) - the amino acids that belong to this class have an affinity for water and tend to absorb it.

The HP model is based on the observation that the hydrophobic forces are very important factors in the protein folding process, guiding the protein to its native three dimensional structure.

The primary structure of a protein is seen as a sequence of n amino acids and each amino acid is classified in one of the two categories: hydrophobic (H) or hydrophilic (P). A *conformation* of the protein $\mathcal{P}$ is a function C, that maps the protein sequence $\mathcal{P}$ to the points of a two-dimensional cartesian lattice such that any two consecutive amino acids in the primary structure of the protein are neighbors (horizontally or vertically) in the bidimensional lattice. It is considered that any position of an amino acid in the lattice may have a maximum number of 4 neighbors: up, down, left, right.

A configuration C is *valid* if it is a *self avoiding path*, i.e the mapped positions of two different amino acids must not be superposed in the lattice.

Figure 1 shows a configuration example for the protein sequence $\mathcal{P} = HHPH$, of length 4, where the hydrophobic amino acids are represented in black and the hydrophilic ones are in white.

The energy function in the HP model reflects the fact that hydrophobic amino acids have a propensity to form a hydrophobic core. Consequently the energy function adds a value of -1 for each two hydrophobic amino acids

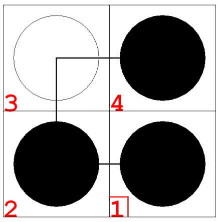

Figure 1. A protein configuration for the sequence $\mathcal{P} = HHPH$, of length 4. Black circles represent hydrophobic amino acids, while white circles represent hydrophilic ones. The configuration may be represented by the sequence LUR. The value of the energy function for this configuration is -1.

that are mapped by C on neighboring positions in the lattice, but that are not consecutive in the primary structure $\mathcal{P}$. Such two amino acids are called topological neighbors. Any hydrophobic amino acid in a valid conformation C can have at most 2 such neighbors (except for the first and last aminoacids, that can have at most 3 topological neighbors).

The computational protein folding problem in the HP model is to find the conformation C whose energy is minimum. A solution for the bidimensional HP protein folding problem, corresponding to an n-length sequence P could be represented by a $n-1$ length sequence $\pi = \pi_1\pi_2...\pi_{n-1}$, $\pi_i \in \{L, R, U, D\}$, $\forall 1 \leq i \leq n-1$, where each position encodes the direction of the current amino acid relative to the previous one (L-left, R-right, U-up, D-down). As an example, the solution configuration corresponding to the sequence presented in Figure 1 is LUR.

3. A Reinforcement Learning Model for Solving the Bidimensional Protein Structure Prediction Problem

In the folowing we are addressing the *Bidimensional Protein Structure Prediction* problem ($BPSP$), more exactly the problem of predicting the bidimensional structure of proteins, but our model can be easily extended to the three-dimensional protein folding problem.

Let us consider, in the following, that $\mathcal{P} = p_1p_2...p_n$ ($n \geq 3$) is a protein HP sequence consisting of n amino acids, where $p_i \in \{H, P\}$, $\forall 1 \leq i \leq n$. As we have indicated in Section 2, the bidimensional structure of $\mathcal{P}$ will be represented as an $n-1$-dimensional sequence $\pi = \pi_1\pi_2...\pi_{n-1}$, where each element π_k $(1 \leq k \leq n)$ encodes the direction (L, U, R or D) of the current amino acid location relative to the previous one.

The RL task associated to the $BPSP$ is defined as follows.

The state space $\mathcal{S}$ (the agent's environment) will consist of $\frac{4^n-1}{3}$ states, i.e $\mathcal{S} = \{s_1, s_2, ..., s_{\frac{4^n-1}{3}}\}$. The *initial state* of the agent in the environment is s_1. A state $s_{i_k} \in \mathcal{S}(i_k \in [1, \frac{4^n-1}{3}])$ reached by the agent at a given moment after it has visited states $s_1, s_{i_1}, s_{i_2}, ...s_{i_{k-1}}$ is a *terminal* (final or goal) state if the number of states visited by the agent in the current sequence is $n-1$, i.e. $k = n-2$. A path from the initial to a final state will represent a possible bidimensional structure of the protein sequence $\mathcal{P}$.

The action space $\mathcal{A}$ consists of 4 actions available to the problem solving agent and corresponding to the 4 possible directions $L(Left)$, $U(Up)$, $R(Right)$, $D(Down)$ used to encode a solution, i.e $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, where $a_1 = L$, $a_2 = U$, $a_3 = R$ and $a_4 = D$.

The transition function $\delta : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ between the states is defined as in Formula 1.

(1)
$$\delta(s_{\frac{4^k-1}{3}+i}, a_l) = s_{\frac{4^{k+1}-1}{3}+4\cdot(i-1)+l} \; \forall k \in [0, n-1], \; \forall i, 1 \le i \le 4^k \; \forall l, 1 \le l \le 4.$$

This means that, at a given moment, from a state $s \in \mathcal{S}$ the agent can move in 4 successor states, by executing one of the 4 possible actions. We say that a state $s' \in \mathcal{S}$ that is accessible from state s, i.e $s' \in \bigcup_{a \in \mathcal{A}} \delta(s, a)$, is the *neighbor* (*successor*) state of s.

The transitions between the states are equiprobable, the transition probability $P(s, s')$ between a state s and each neighbor state s' of s is equal to 0.25 .

Let us consider a path π in the above defined evironment from the initial to a final state, $\pi = (\pi_0 \pi_1 \pi_2 \cdots \pi_{n-1})$, where $\pi_0 = s_1$ and $\forall 0 \le k \le n-2$ the state π_{k+1} is a *neighbor* of state π_k. The sequence of actions obtained following the transitions between the successive states from path π will be denoted by $a_\pi = (a_{\pi_0} a_{\pi_1} a_{\pi_2} \cdots a_{\pi_{n-2}})$, where $\pi_{k+1} = \delta(\pi_k, a_{\pi_k})$, $\forall 0 \le k \le n-2$. The sequence a_π will be refered as the *configuration* associated to the path π and it can be viewed as a possible bidimensional structure of the protein sequence $\mathcal{P}$. Consequently we can associate to a path π a value denoted by E_π representing the energy of the bidimensional configuration a_π of protein $\mathcal{P}$ (Section 2).

The *BPSP* formulated as a RL problem will consist in training the agent to find a path π from the initial to a final state that will corespond to the bidimensional structure of protein $\mathcal{P}$ given by the coresponding configuration a_π and having the minimum associated energy.

It is known that the estimated utility of a state [6] in a reinforcement learning process is the estimated *reward-to-go* of the state (the sum of rewards received from the given state to a final state). So, after a reinforcement learning process, the agent learns to execute those transitions that maximize the sum of rewards received on a path from the initial to a final state.

As we aim at obtaining a path π having the minimum associated energy E_π, we define the reinforcement function as follows: if the transition generates a configuration that is not *valid* (i.e self-avoiding) (see Section 2) the received reward is 0.01; the reward received after a transition to a non terminal state is a small positive constant greater than 0.01 (e.q 0.1); the reward received after a transition to a final state π_{n-1} after states $s_1, \pi_1, \pi_2, ...\pi_{n-2}$ were visited is minus the energy of the bidimensional structure of protein $\mathcal{P}$ corresponding to the configuration a_π.

Considering the reward defined as indicated above, as the learning goal is to maximize the total amount of rewards received on a path from the initial to a final state, it can be easily shown that the agent is trained to find a self avoiding path π that minimizes the associated energy E_π.

3.1. **The learning process.**

During the training step of the learning process, the agent will determine its *optimal policy* in the environment, i.e the *policy* that maximizes the sum of the received rewards.

For training the *BPSP* agent, we propose a Q-learning approach. The idea of the training process is the following:

- The Q values are initialized with 0.
- During some training episodes, the agent will experiment (using the ϵ-Greedy action selection mechanism) some (possible optimal) paths from the initial to a final state, updating the Q-values estimations according to the $Q - learning$ algorithm [7].
- During the training process, the Q-values estimations converge to their exact values, thus, at the end of the training process, the estimations will be in the vicinity of the exact values.

After the training step of the agent has been completed, the solution learned by the agent is constructed by starting from the initial state and following the *Greedy* mechanism until a solution is reached. From a given state i, using the *Greedy* policy, the agent transitions to a neighbor j of i having the maximum Q-value. Consequently, the solution of the *BPSP* reported by the RL agent is a path $\pi = (s_1\pi_1\pi_2\cdots\pi_{n-2})$ from the initial to a final state, obtained following the policy described above. We mention that there may be more than one optimal policy in the environment determined following the *Greedy* mechanism described above. In this case, the agent may report a single optimal policy of all optimal policies, according to the way it was designed.

It is proven in [8] that the Q-values learned converge to their optimal values as long as all state-action pairs are visited an infinite number of times. Consequently, the configuration a_π corresponding to the path π learned by the $BPSP$ agent converges, in the limit, to the sequence that corresponds to the bidimensional structure of protein $\mathcal{P}$ having the minimum associated energy.

4. Experiment

In this section we aim at experimentally evaluating the proposed *reinforcement learning* approach.

Let us consider a bidimensional HP protein instance $\mathcal{P} = HPHPPHHP HPPHPHHPPHPH$, consisting of twenty amino acids, i.e $n = 20$. The benchmark instance for the 2D HP Protein Folding Problem used in this study can be found in [9] and its known optimal energy value is $E^* = -9$. As we have presented in Section 3, the states space will consist of $\frac{4^{20}-1}{3}$ states. We have trained the $BPSP$ agent as indicated in Subsection 3.1. As proven in [8], the Q-learning algorithm converges to the optimal Q-values as long as all state-action pairs are visited an infinite number of times, the learning rate α is small (e.q 0.01) and the policy converges in the limit to the Greedy policy. We remark the following regarding the parameters setting:

- the learning rate is $\alpha = 0.01$ in order to assure the convergence of the algorithm;
- the discount factor for the future rewards is $\gamma = 0.9$;
- the number of training episodes is $19 \cdot 10^5$;
- the ϵ-Greeedy action selection mechanism was used. Regarding the ϵ parameter used for the *epsilon*-Greedy action selection mechanism during the training step, the following strategy was used: we have started with $\epsilon = 1$ in order to favor exploration, then after the training progresses ϵ is decreased until it reaches a small value, which means that at the end of the training exploitation is favorized.

Using the above defined parameters and under the assumptions that the state action pairs are equally visited during training, the solution reported after the training of the $BPSP$ agent was completed is the *configuration* $a_\pi = (RUULDLULLDRDRDLDRRU)$, determined starting from state s_1, following the *Greedy* policy (as we have indicated in Subsection 3).

The solution learned by the agent is represented in Figure 2 and has an energy of -9.

Consequently, the $BPSP$ agent learns the optimal solution of the computational bidimensional protein folding problem, i.e the bidimensional structure of the protein $\mathcal{P}$ that has a minimum associated energy (-9).

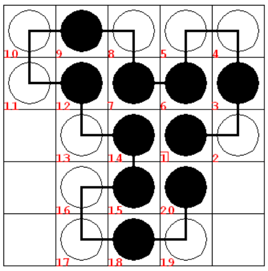

FIGURE 2. The learned solution is *RUULDLULLDRDRDLDRRU*. The value of the energy function for this configuration is -9.

5. Comparison with related approaches

Regarding the Q-learning approach introduced in Section 3 for solving the bidimensional protein folding problem, we remark the following. The training process during an episode has a time complexity of $\theta(n)$, where n is the length of the HP protein sequence. Consequently, assuming that the number of training episodes is k, the overall complexity of the algorithm for training the $BPSP$ agent is $\theta(k \cdot n)$.

In the following we will briefly compare our approach with some of the existing approaches. The comparison is made considering the computational time complexity point of view. Since for the most of the existing approaches the authors do not provide the asymptotic analysis of the time complexity of the proposed approaches, we can not provide a detailed comparison.

Genetic and evolutionary approaches were developed in [10, 11, 12] for predicting the bidimensional structure of proteins. An asymptotic analysis of the computational complexity for evolutionary algorithms (EAs) is difficult [13] and is usually done only for particular problems. Anyway, the number of generations (or equivalently the number of fitness evaluations) is the most important factor in determining the order of EA's computation time. In our view, the time complexity of an evolutionary approach for solving the problem of predicting the structure of an n-dimensional protein is at least $noOfRuns \cdot n \cdot noOfGenerations \cdot populationLength$. For large instances, it is likely (even if we can not rigurously prove) that the computational complexity of our approach is less than the one of an evolutionary approach.

Ant Colony Optimization (ACO) was already used for solving the protein folding problem in the HP model [14, 15]. Neumann et al. show in [16] how simple ACO algorithms can be analyzed with respect to their computational complexity on example functions with different properties, and also claim that asymptotic analysis for general ACO systems is difficult. In our view, the time complexity of an ACO approach for solving the problem of predicting th structure of an n-dimensional protein is at least $noOfRuns \cdot n \cdot noOfIterations \cdot noOfAnts$. For large instances, it is likely (even if we can not rigurously prove) that our approach has a lower computational complexity.

Compared to the supervised classification approach from [17], the advantage of our RL model is that the learning process needs no external supervision, as in our approach the solution is learned from the rewards obtained by the agent during its training. It is well known that the main drawback of supervised learning models is that a set of inputs with their target outputs is required, and this can be a problem.

The main drawback of our approach is that a very large number of training episodes has to be considered in order to obtain accurate results and this leads to a slow convergence. In order to speed up the convergence process, further improvements, such as local search mechanisms will be considered. Anyway, we think that the direction of using reinforcement learning techniques in solving the protein folding problem is worth being studied and further improvements can lead to valuable results.

6. Conclusions and Further Work

We have proposed in this paper a reinforcement learning based model for solving the bidimensional protein structure prediction problem, a fundamental problem in computational molecular biology and biochemical physics. To our knowledge, except for the ant based approaches, the problem of predicting the bidimensional structure of proteins has not been addressed in the literature using reinforcement learning, so far. The model proposed in this paper can be easily extended to solve the three-dimensional computational protein folding problem, and moreover to solve other optimization problems.

We plan to extend the evaluation of the proposed RL model for other large HP protein sequences, to further test its performance. We will also investigate possible improvements of the RL model by analyzing a temporal difference approach [1], by using different reinforcement functions and by adding different local search mechanisms in order to increase the model's performance. An extension of the $BPSP$ model to a distributed RL approach will be also considered.

ACKNOWLEDGEMENT

This work was supported by CNCSIS - UEFISCDI, project number PNII - IDEI 2286/2008.

References

[1] Sutton, R.S., Barto, A.G., *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[2] Dorigo, M., Stützle, T., *Ant Colony Optimization*, Bradford Company, Scituate, MA, USA, 2004.

[3] Dill, K.A.A., Ozkan, S.B.B., Weikl, T.R.R., Chodera, J.D.D., Voelz, V.A.A., *The protein folding problem: when will it be solved?*, Current Opinion in Structural Biology **17**, 2007, pp. 342-346.

[4] Anfinsen, C.B., *Principles that govern the folding of protein chains*, Science **181**, 1973, pp. 223–230.

[5] Dill, K., Lau, K., *A lattice statistical mechanics model of the conformational sequence spaces of proteins*, Macromolecules **22**, 1989, pp. 3986–3997.

[6] Russell, S., Norvig, P., *Artificial Intelligence - A Modern Approach*, Prentice Hall International Series in Artificial Intelligence, Prentice Hall, 2003.

[7] Dayan, P., Sejnowski, T., *TD(lambda) converges with probability 1*, Mach. Learn. **14**, 1994, pp. 295–301.

[8] Watkins, C.J.C.H., Dayan, P., *Q-learning*, Machine Learning **8**, 1992, pp. 279–292.

[9] Hart, W., Istrail, S., *HP benchmarks* http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html.

[10] Unger, R., Moult, J., *Genetic algorithms for protein folding simulations*, Mol. Biol. **231**, 1993, pp. 75–81.

[11] Zhang, X., Wang, T., Luo, H., Yang, Y., Deng, Y., Tang, J., Yang, M.Q., *3D protein structure prediction with genetic Tabu search algorithm*, BMC Systems Biology **4**, 2009, pp. 1–9.

[12] Chira, C., *Hill-climbing search in evolutionary models for protein folding simulations*, Studia **LV**, 2010, pp. 29–40.

[13] Hart, W.E., Belew, R.K., *Optimizing an arbitrary function is hard for the genetic algorithm*, In: Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, 1991, pp. 190–195.

[14] Shmygelska, A., Hoos, H., *An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem*, BMC Bioinformatics **6**, 2005, pp. 1–22.

[15] Thalheim, T., Merkle, D., Middendorf, M., *Protein folding in the HP-model solved with a hybrid population based aco algorithm*, IAENG International Jurnal of Computer Science **35**, 2008, pp. 1–10.

[16] Neumann, F., Sudholt, D., Witt, C., *Computational complexity of ant colony optimization and its hybridization with local search.* In: Innovations in Swarm Intelligence, 2009, pp. 91–120.

[17] Ding, C.H.Q., Dubchak, I., *Multi-class protein fold recognition using support vector machines and neural networks*, Bioinformatics **17**, 2001, pp. 349–358.

Babeş-Bolyai University, Department of Computer Science, 1, M. Kogălniceanu street, 400084 Cluj-Napoca, Romania

E-mail address: {gabis,iuliana,istvanc}@cs.ubbcluj.ro