Protein Folding in Simplified Models With Estimation of Distribution Algorithms

Roberto Santana, Pedro Larrañaga, and Jose A. Lozano, *Member, IEEE*

***Abstract*—Simplified lattice models have played an important role in protein structure prediction and protein folding problems. These models can be useful for an initial approximation of the protein structure, and for the investigation of the dynamics that govern the protein folding process. Estimation of distribution algorithms (EDAs) are efficient evolutionary algorithms that can learn and exploit the search space regularities in the form of probabilistic dependencies. This paper introduces the application of different variants of EDAs to the solution of the protein structure prediction problem in simplified models, and proposes their use as a simulation tool for the analysis of the protein folding process. We develop new ideas for the application of EDAs to the bidimensional and tridimensional (2-d and 3-d) simplified protein folding problems. This paper analyzes the rationale behind the application of EDAs to these problems, and elucidates the relationship between our proposal and other population-based approaches proposed for the protein folding problem. We argue that EDAs are an efficient alternative for many instances of the protein structure prediction problem and are indeed appropriate for a theoretical analysis of search procedures in lattice models. All the algorithms introduced are tested on a set of difficult 2-d and 3-d instances from lattice models. Some of the results obtained with EDAs are superior to the ones obtained with other well-known population-based optimization algorithms.**

***Index Terms*—Estimation of distribution algorithm (EDAs), hydrophobic-polar (HP) model, protein folding, protein structure prediction.**

I. Introduction

PROTEINS play a fundamental role in nature. These structures made of amino acids participate in many important tasks that guarantee the correct functioning of living cells. The protein structure is the result of the so-called protein folding process in which the initially unfolded chain of amino acids is transformed into its final structure. Under suitable conditions, this structure is uniquely determined by the sequence.

Manuscript received March 21, 2007; revised June 12, 2007. Published July 30, 2008 (projected). The work of J. A. Lozano was supported by Grant PR2006-0315, while at the University of California, San Diego. This work was supported in part by SAIOTEK-Autoinmune (II) 2006 Research Project from the Basque Government, and in part by the Etortek Research Project from the Basque Government, in part by the Spanish Ministerio de Ciencia y Tecnología under Grant TIN 2005–03824, and in part by the SGI/IZO-SGIker UPV/EHU (supported by the Spanish Program for the Promotion of Human Resources within the National Plan of Scientific Research, Development and Innovation-Fondo Social Europeo and MC y T), gratefully acknowledged for generous allocation of computational resources.

R. Santana and J. A. Lozano are with the Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 20080 San Sebastian–Donostia, Guipuzcoa, Spain (e-mail: rsantana@si.ehu.es; ja.lozano@ehu.es).

P. Larrañaga is with the Department of Artificial Intelligence, Technical University of Madrid, 28660 Boadilla del Monte, Madrid, Spain (e-mail: pedro.larranaga@fi.upm.es).

Digital Object Identifier 10.1109/TEVC.2007.906095

The complex and challenging nature of the protein folding process is highlighted by the Levinthal paradox [1]. If an unfolded protein were to attain its corrected folded configuration by sequentially sampling all possible configurations, then it would require a huge time exponential in the number of residues. This happens because each molecule could adopt an astronomical number of configurations. Nevertheless, proteins in nature can fold very quickly, often within a matter of seconds. Obviously, the protein folding does not proceed as an exhaustive search. Therefore, it is fundamental to understand how proteins attain their native configuration. However, the exact laws that govern the protein folding process are unknown, and the problem of finding the 3-d native structure of the protein given its sequence of amino acids is open.

Protein modeling and the computational simulation of protein mechanisms have proved to be valuable tools to answer the questions posed in the biological domain. Since it is difficult to scale modeling to a fine level of detail, some simplified models have been proposed in order to study, to different extents, the protein folding process. In this paper, we concentrate on a class of coarse-grained models that have been extensively used to study approximations of the protein folding problem. Using this model, we propose the use of estimation of distribution algorithms (EDAs) for two related problems: to find the native structure of the protein from its sequence, and to simulate the protein folding mechanism.

Most of the application results for EDAs on discrete problems have been achieved for problems with binary representation. Theoretical analysis of the EDA behavior for discrete problems is also mainly constrained to problems with binary representation. The use of protein coarse-grained models as a testbed for EDAs can help to advance the understanding of EDAs and to investigate their performance when applied to nonbinary problems. Coarse-grained models have also been 6treated with a variety of optimization methods, allowing us to evaluate the performance of EDAs in comparison with these algorithms.

The protein model of choice is the hydrophobic-polar (HP) model [2], which is based on the fact that hydrophobic interactions are a dominant force in protein folding. The HP model has arisen as a suitable benchmark for cross-disciplinary studies involving domains such as computational biology, statistical and chemical physics, and optimization. This research has revealed different but related facets of the protein folding problem. In computational biology, the HP model has served to study sequence-structure mapping in proteins [3], to analyze the role of local structures in protein folding [4], and to study aspects related to protein design [5]. In statistical and chemical physics, the HP model has been used for exhaustive generation and analysis of protein conformations [6]. More recently, it has also

been used for doing folding and unfolding simulations [7] and to study density and ground states of the protein folding model [8], [9].

In the optimization domain, the search for the protein structure is transformed into the search for the optimal configuration given an energy function that takes into account the HP interactions that arise in the model. The problem of finding such a minimum energy configuration is NP-complete for the bidimensional (2-d)[10] and tridimensional (3-d) [11] lattices. Performance-guaranteed approximation algorithms of bounded complexity have been proposed to solve this problem [12], but the error bound guaranteed is not small enough for many applications.

The number of optimization heuristics applied to the HP model is extensive [13]–[19] with a significant number of the contributions made in recent years [20]–[30]. Work on evolutionary search applied to protein structure prediction and protein folding for lattice models and real proteins has been surveyed in [31]. A well documented review of current approaches to protein structure prediction is provided in [32].

In an early and very influential paper [19], Unger and Moult described a genetic algorithm (GA) application that used heuristic-based crossover and mutation operators to solve the HP model. The GA was able to outperform a number of variants of Monte Carlo methods at different sequences. Remarkably, the authors identified GAs as being particularly suited to reproducing some aspects of the protein folding process.

Although other GAs have been proposed to address the problem of structure prediction in the HP model, the difficulties of crossover operators to deal with this type of problem have been acknowledged [16], [33]. Particularly, it has been pointed out that one-point and uniform crossover do not perform well for this problem, and a number of explanations for this have been proposed [33]. Even if the GA results have been shown to improve by employing more sophisticated operators, another more general alternative is the conception of evolutionary algorithms able to learn and use the relevant interactions that may arise between the variables of the problem.

EDAs [34]–[37] are evolutionary algorithms that construct an explicit probability model of a set of selected solutions. This model can capture, by means of probabilistic dependencies, relevant interactions among the variables of the problem. The model can be conveniently used to generate new promising solutions. In [27], an EDA that uses a Markov probabilistic model outperformed other population-based methods when solving the 2-d protein folding problem. In the present paper, we further improve and generalize the results achieved in [27] by considering other types of probabilistic models, and by treating the more challenging class of 3-d simplified protein folding models. On the other hand, we investigate the use of EDAs as a simulating tool for the protein folding mechanism. Starting from current biological approaches to the protein folding process, and based on EDA capabilities to save and update probability models of the search, we provide evidence showing that EDAs can mimic the protein folding process to a certain extent, and more consistently than GAs.

This paper is arranged as follows. In Section II, we briefly review the biological concepts related to protein folding, and introduce the HP model and the functional model protein. Section III reviews a number of previous approaches to the solution of simplified models using evolutionary and Monte Carlo-based algorithms. Section IV presents the class of EDAs. Section V introduces the problem representation and discusses how the probability model can capture the regularities that may arise in the HP problem. In Section VI, the probabilistic models and the EDAs used for the protein structure problem are introduced. This section also presents the EDA model of protein folding. In Section VII, the experimental benchmark is introduced and numerical results of our experiments are presented. Finally, in Section VIII, the conclusions of our research are given, and further work is discussed.

II. Protein Folding

We will briefly recall some of the main biological concepts related to the protein folding problem that are relevant to our discussion.

Proteins are macromolecules made out of 20 different amino acids, also referred to as residues. An amino acid has a peptide backbone and a distinctive side chain group. The peptide bond is defined by an amino group and a carboxyl group connected to an alpha carbon to which a hydrogen and side chain group are attached.

Amino acids are combined to form sequences which are considered the primary structure of the peptides or proteins. The secondary structure is the locally ordered structure brought about via hydrogen bounding mainly within the peptide backbone. The most common secondary structure elements in proteins are the alpha helix and the beta sheet. The tertiary structure is the global folding of a single polypeptide chain.

Under specific conditions, the protein sequence folds into a unique native 3-d structure. Each possible protein fold has an associated energy. The *thermodynamic hypothesis* states that the native structure of a protein is the one for which the free energy achieves the global minimum. Based on this hypothesis, many methods that search for the protein native structure define an approximation of the protein energy and use optimization algorithms that look for the protein fold that minimizes this energy. These approaches mainly differ in the type of energy approximation employed and in the characteristics of the protein modeling.

The achievement of the protein native structure is the result of the so-called protein folding process. The laws that govern protein folding are unknown. Therefore a number of ideas have emerged that try to answer this question: how do amino acid sequences specify proteins 3-d structure?

There are two main approaches to protein folding, commonly referred as the "classical" and "new" views. The "classical" view considers folding as a defined sequence of states leading from the unfolded to the native state. This sequence is called the pathway [38]. In the "new" view approach, folding is seen as the progressive organization of an ensemble of partially folded structures through which the protein passes on its way to the folded structure [39]. This approach emphasizes the idea of each state being an ensemble of rapidly interconverting conformations. One of the main differences between both approaches is that the "new" view allows for a more heterogeneous transition

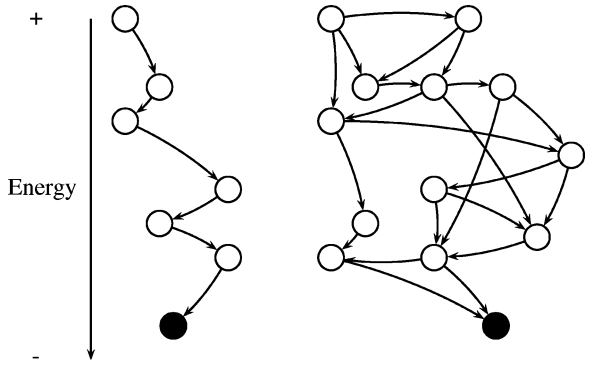


Fig. 1. Schematic representation of the "classical" (left) and "new" (right) views of protein folding.

state than the "classical" view, which concentrates on a single, well-defined folding pathway [40].

Fig. 1 shows one schematic representation of the "classical" (left) and "new" (right) views of protein folding. In the figure, each possible protein configuration is represented as a circle, and arrows represent possible transitions between configurations. In both approaches, the native state (filled circle) is achieved when the energy is minimized.

A. The "New" View of Protein Folding

The EDA-based model of protein folding presented in this paper adopts the second view. Therefore, we must study in greater detail some of the aspects related to it.

In the "new" view approach, the *energy landscape of a folding protein* resembles a partially rough funnel. The local roughness of the funnel reflects transient trapping of the protein configurations in local free energy minima. Another important role is played by *frustration*.

There are two main sources of frustration in a protein [41]: energetical and topological. We will focus on energetical frustration which is associated with the amino acid sequence in the protein. It occurs when incorrect contacts are formed as the chain folds, when the sequence forces mismatched residues to be in contact in the native state or when there is competition between the protein interactions (i.e., not all of them can be simultaneously satisfied). The importance of frustration due to competing interactions and its influence in the emergence of highly multimodal search landscapes, which are difficult to optimize using GAs, have been studied in [42].

The emergence of frustration and other properties of proteins can be analyzed by using *order parameters* or *progress coordinates* which help to describe and quantify the protein ensembles during the protein folding process. They are used to explore the connection between the folding process and the topology of the protein native state. A parallel can be traced between the role of order parameters in protein folding and the role of parameters commonly used to describe the behavior of evolutionary algorithms (e.g., average fitness of the population at each generation, convergence and diversity measures, etc.) for functions of different difficulty.

Examples of order parameters are the *contact order* and the *volume of the protein*. Another quantifying measure is the *folding rate*. The *contact order* of the protein is the average sequence separation between residues that make contact in the three-dimensional structure. The *volume* is a measure of the degree of folding of the protein, allowing distinction between compact and extended conformations. The *folding rate* is the amount of time the protein takes to fold.

In the case of small proteins, other measurements of the folding reactions can be made. Among them are the distribution of structures in the transition state ensemble, and the structure of the native state. The *fraction of native contacts* Q that exist in the current conformation [43] can be used as a measure of frustration. For a given conformation, Q varies between 0 and 1, with the native conformation at $Q = 1$. It is also possible to compute the total free energy $F_{\text{tot}}(Q)$ as a function of Q

$$F_{\text{tot}}(Q) = F_{\text{int}}(Q) - T \cdot S_{\text{conf}}(Q) \tag{1}$$

where $F_{\text{int}}(Q)$ is the average internal energy of conformations with Q native contacts, T is the temperature of the system, and $S_{\text{conf}}(Q)$ is the corresponding conformational entropy (the logarithm of the number of accessible conformations with Q native contacts) [40].

We enumerate a number of facts commonly accepted and explained in the "new" view of protein folding [39], [40], [43], [44]. Some of these issues will be investigated through simulation of the EDA-based model.

- The folding rates of small proteins correlate with their contact order. Proteins with a large fraction of their contacts between residues close in sequence tend to fold faster than proteins with more nonlocal interactions.
- Protein folding rates and mechanisms are largely determined by the protein native topology. Proteins with similar native states are expected to exhibit a similar protein folding behavior.
- Local interactions are more likely to form early in folding than nonlocal interactions.
- During the folding process, the energy of the structures will decrease on average as they become more and more similar to the native structure of a natural protein.
- Folding is not only determined by properties of the folded state but also by the energetic difference between the folded and unfolded ensembles of states.
- The geometrical accessibility of different native contacts is different, and therefore some are more easily formed than others.
- Some contacts may be topologically required (or at least be more likely) to be formed before others during folding.

B. The HP and Functional Model Protein

This section briefly introduces the HP and functional model protein.

The HP model considers two types of residues: hydrophobic (H) residues and hydrophilic or polar (P) residues. A protein is considered a sequence of these two types of residues, which are located in regular lattice models forming self-avoided paths. Given a pair of residues, they are considered neighbors if they are adjacent either in the chain (connected neighbors) or in the lattice but not connected in the chain (topological neighbors).

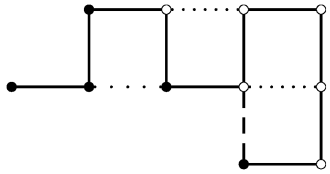


Fig. 2. One possible configuration of sequence $HHHPHPPPPPH$ in the HP model. There is one HH (represented by a dotted line with wide spaces), one HP (represented by a dashed line) and two PP (represented by dotted lines) contacts.

The total number of topological neighboring positions in the lattice (z) is called the lattice coordination number.

For the HP model, an energy function that measures the interaction between topological neighbor residues is defined as $\epsilon_{HH} = -1$ and $\epsilon_{HP} = \epsilon_{PP} = 0$. The HP problem consists of finding the solution that minimizes the total energy. In the linear representation of the sequence, hydrophobic residues are represented with the letter H and polar ones with P. In the graphical representation, hydrophobic proteins are represented by black beads and polar proteins, by white beads. Fig. 2 shows the graphical representation of a possible configuration for sequence $HHHPHPPPPPH$. The energy that the HP model associates with this configuration is -1 because there is only one HH contact, arisen between the second and fifth residues.

Although more complex models have been proposed [45]–[48], the HP model remains a focus of research in computational biology [3], [5], [6], [49] and chemical and statistical physics [7]–[9]. In evolutionary computation [17], [22], [26], [27], [29], [33], [50], the model is still employed given its simplicity and its usefulness as a testbed for new evolutionary optimization approaches.

The functional model protein is a "shifted" HP model. The name comes from the fact that the model supports a significant number of proteins that can be characterized as functional. This model has native states, some of which are not maximally compact. Thus, in some cases, they have cavities or potential binding sites, a key property that is required in order to investigate ligand binding using these models [51]. The energy values associated with the model contain both attractive $\epsilon_{HH} = -2$ and repulsive interactions ($\epsilon_{PP} = 1$, $\epsilon_{HP} = 1$, and $\epsilon_{PH} = 1$). Again, the objective is to minimize the total energy. For example, the energy that the functional model protein associates with the configuration shown in Fig. 2 is 1 because there is one HH (represented by a wide dotted line), one HP (represented by a dashed line) and two PP (represented by dotted lines) contacts.

III. Review of Previous Evolutionary Methods

Previous population-based approaches to simplified protein folding include the use of nature-inspired and Monte Carlo methods [52]. Some versions of these methods are compared with EDAs in the experiments section.

Since the publication of the Unger and Moult paper [19] on the use of GAs for protein structure prediction, new issues have arisen along with new points of view on the protein folding problem. However, GA applications to the HP problem are many and varied. In [15], a search strategy called pioneer search was used together with a simple GA. Although the algorithm improved some of the results achieved in [19], it was unable to find the optimal solutions for the longest instances considered. In [30], a GA with specialized genetic operators was used to solve HP sequences up to 50 residues. The algorithm found the best solutions for the six sequences tried, but increasing the GA efficiency was acknowledged to be a requirement for solving longer sequences.

In [21] and [53], evolutionary algorithms for the 3-d HP problem are proposed. While in [53] a simple GA showed no better results than those achieved in [19], a more sophisticated approach is presented in [21]. By using a backtracking-based repairing procedure, the latter algorithm guarantees that the search is constrained to the space of legal solutions. Since the number of self-avoided paths on square lattices is exponential in the length of the sequence [19], generating legal solutions with a backtracking algorithm is a feasible alternative.

The multimeme algorithm (MMA) for protein structure prediction [17] is a GA combined with a set of local searches. From this set, the algorithm self-adaptively selects which local search heuristic to use for different instances, states of the search, or individuals in the population. This algorithm was used to find solutions of the functional model protein. A relevant issue of this algorithm is the use of a contact map memory as a way to collect and use important problem information. Contact maps abstract the geometric details of the structures, keeping only the essential topological features of the configurations. In [26], MMA was extended by the incorporation of fuzzy-logic-based local searchers. The modifications led to obtain a more robust algorithm that improved previous MMA results in the protein structure prediction problem. Memetic algorithms were also combined with a population of rules [29] to solve the HP model in a two-dimensional triangular lattice. The algorithm proposed outperformed simple versions of GAs and memetic algorithms.

Different variants of immune algorithms (IAs) [22], [23], [50] have been proposed for the HP problem. These evolutionary algorithms inspired in the theory of clonal selection, use hypermacromutation and aging as important operators to proceed the search. In [50], the algorithm found the optimal configurations of the regular 2-d HP model for the smallest problems. The algorithm failed to find the optimum for the longest instances. In [22], the original IA is developed to include a memory mechanism that improves results for the 2-d regular lattice. Recently, IA has been used with very good results to solve HP problems on the 3-d lattice and instances from the functional protein model [23].

Traditional Monte Carlo methods that use Markov chains sample from the protein folding space one point at a time. Due to the rugged landscape, these methods tend to get trapped in local minima. New Monte Carlo methods have been proposed to cope with these problems [54]. Among the alternatives proposed, two main classes of the strategies used by the Monte Carlo methods can be distinguished: to use chain growth algorithms [13], or to sample the space with a population of Markov chains in which a different temperature is attached to each chain [55]. Chain growth algorithms [25], [56]–[58] such as the pruned-enriched Rosenbluth method (PERM) [25] are based on growing the sequence conformation by successively adding individual particles, guiding the growth towards configurations with lower energies, and using population control to eliminate bad configurations and increase good ones [58].

Although chain growth methods have achieved some of the best results for HP models in regular 2-d and 3-d lattices, it is important to emphasize that algorithms such as PERM combine the process of constructing the solutions with the evaluation step of the solutions. Even if this strategy allows one to use more information about the HP fitness function, it lacks generality because it cannot be applied to problems where subsolutions cannot be independently evaluated.

All the above-mentioned algorithms either use genetic operators or Markov chain transitions. They do not use any model of the search space. An algorithm that incorporates, to a certain scale, the modeling step is the ant colony optimization (ACO) method presented in [18] and [28]. In this approach, the simulated ants construct candidate conformations for a given HP protein sequence, apply a local search to achieve further improvement, and update a probability value based on the quality of the solutions found. In ACO terminology, this value is called the pheromone trail.

Additionally, we mention that there are several examples of the application of GAs and other evolutionary algorithms to protein structure prediction problems using more complex protein models. For a review on evolutionary algorithms and other optimization methods applications to protein problems, [59], [60] can be consulted. We briefly analyze some connections between the use of evolutionary algorithms for simple and more complex protein models.

While more complex protein folding models (e.g., rotamer-based protein models [45]) can provide more realistic results in some protein folding and protein design studies, they are of limited use for other tasks (e.g., the exhaustive generation and analysis of protein configurations for many instances). Similarly, there are successful applications of EDAs to rotamer-based protein models [61], [62], but the complexity of these models would not allow one to conduct the sort of detailed analysis that will be presented in Section V. To summarize, the decision of using the HP or more complex protein models will depend on the type of optimization or simulation tasks addressed.

However, the application of GAs to more complex models can provide clues for the simulations done with simpler models. In [63], an evolutionary algorithm is applied to de novo all-atom folding of a protein comprising 60 aminoacids. The authors use the concept of "native content" of the population of solutions to evaluate convergence of the algorithm. A related idea will be presented in Section VII-E1, where we use the number of native contacts to investigate the way in which our evolutionary algorithm samples the energy landscape of the function.

In [64], a GA is used for protein structure prediction of 28 fragments of protein structures up to 14 residues long. The search is done in the torsion space of the protein atoms. The GA protocol is successful in finding a lower energy than the corresponding minimized structure in 26 out of the 28 cases examined. The high computational cost of the algorithm is attributed by the authors to the difficulty of finding acceptable crossover conformations. We notice that one of the advantages of EDAs is their ability to avoid the disruption caused in the structure of the solutions by crossover operators.

Results achieved in [62]–[64] also show that in the case of protein structure prediction with complex models the addition of local optimization techniques to GAs and EDAs may be a requirement for attaining realistic protein conformations.

Even though this short overview has focused on Monte Carlo and nature inspired methods, we emphasize that there are several applications of heuristic algorithms to the protein structure problem that are beyond the scope of this paper.

IV. Estimation of Distribution Algorithms

The results of the GAs for the protein structure problem can be improved by designing heuristic-based genetic operators. However, the improvements are constrained by the narrow scope of application of these types of knowledge-based operators. A more robust solution is the use of evolutionary algorithms able to use probabilistic models of the search space.

EDAs replace the traditional crossover and mutation operators used in GAs by probabilistic models. These algorithms construct, in each generation, a probabilistic model that estimates the probability distribution of the selected solutions. The probabilistic model must be able to capture, in the form of statistical dependencies, a number of relevant relationships among the variables. Dependencies are then used to generate solutions during a simulation step. It is expected that the generated solutions share a number of characteristics with the selected ones. In this way, the search leads to promising areas of the search space.

EDAs can be seen as a development of GAs. By recombining a subset of selected solutions, GAs are able to process the information learned during the search, and to orient the exploration to promising areas of the search space. Nevertheless, it has been proved that GAs experience limitations in their capacity to deal with problems where there are complex interactions between different components of the solutions [65], [66]. In these scenarios, EDAs can exhibit a better performance. The success of EDAs in the solution of different practical problems has been documented in the literature [34].

The general scheme of the EDA approach is shown in Algorithm 1. The selection method employed can be any of those traditionally used by GAs. In the literature, truncation, Boltzmann, and tournament selection are commonly used with EDAs. A key characteristic and crucial step of EDAs is the construction of the probabilistic model. These models may differ in the order and number of the probabilistic dependencies that they represent.

Algorithm 1: **Main scheme of the EDA approach**

1. $D_0 \leftarrow$ Generate M individuals randomly
2. $l = 1$
3. **do** {
4. $D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to a selection method
5. $p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D^s_{l-1}) \leftarrow$ Estimate the joint probability of selected individuals
6. $D_l \leftarrow$ Sample M individuals (the new population) from $p_l(\boldsymbol{x})$
7. } **until** A stop criterion is met

TABLE I
THE DENSITY OF THE DIFFERENT ENERGY LEVELS HP_f AND FP_f CORRESPONDING TO THE HP AND FUNCTIONAL MODEL PROTEIN OF SEQUENCE $HHHPHPPPPPH$

	$HHHPHPPPPPH$											
$H(\mathbf{x})$	-4	-3	-2	-1	0	1	2	3	4	5	invalid	total
HP_f	0	0	16	1428	9581	0	0	0	0	0	8658	19683
FP_f	2	0	426	490	3407	3376	2020	912	350	42	8658	19683

Different classifications of EDAs can be used to analyze these algorithms. Relevant to our research is the classification according to the complexity of the models used to capture the interdependencies between the variables [67]. Considering methods of learning is done in the probability model, EDAs can be divided into two classes. One class groups the algorithms that do a parametric learning of the probabilities, and the other comprises those algorithms that also undertake structural learning. Parametric and structural learning are also known as model fitting and model selection. To the first class, belong population-based incremental learning (PBIL) [68], compact GA (cGA) [69], the univariate marginal distribution algorithm (UMDA) [36], and the factorized distribution algorithm that uses a fixed model of the interactions in all the generations (FDA) [66]. Among EDAs that do a structural learning of the model, are the mutual information maximization for input clustering algorithm (MIMIC) [70], the extended compact GA (EcGA) [71], and EDAs that use Bayesian networks [72]–[74].

V. DEPENDENCIES IN THE SIMPLIFIED PROTEIN MODELS

In this section, we show evidence on the emergence of regularities in the search space of the HP model. In order to achieve this goal, we employ the Boltzmann probability distribution. We start by introducing the problem representation.

A. Problem Representation

Let n be the sequence length. We use X_i to represent a discrete random variable. A possible value of X_i is denoted x_i. Similarly, we use $\mathbf{X} = (X_1, \ldots, X_n)$ to represent an n-dimensional random variable and $\mathbf{x} = (x_1, \ldots, x_n)$ to represent one of its possible values. For a given sequence and lattice, X_i will represent the relative move of residue i in relation to the previous two residues.

Taking as a reference the location of the previous two residues in the lattice, X_i takes values in $\{0, 1, \ldots, z-2\}$, where $z-1$ is the number of movements allowed in the given lattice. These values, respectively, mean that the new residue will be located in one of the $z-1$ numbers of possible directions with respect to the previous two locations. Therefore, values for X_1 and X_2 are meaningless. The locations of these two residues are fixed. A solution $\mathbf{x}$ can be seen as a walk in the lattice, representing one possible folding of the protein. The codification used is called relative encoding, and has been experimentally compared with absolute encoding in [16], showing better results.

We use 2-d and 3-d regular lattices. For regular d-dimensional lattices, $z = 2d$, where d is the lattice dimension.

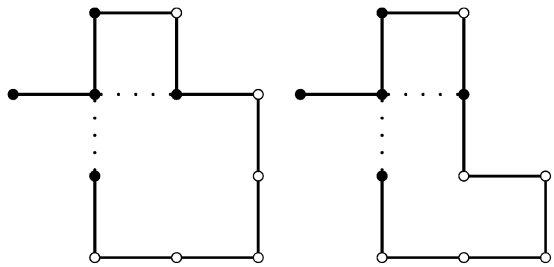


Fig. 3. Best solutions of the functional model protein (left) and HP model (right) for sequence $HHHPHPPPPPH$. The optimal energy values are -4 and -2 for the functional and HP model, respectively. HH contacts are shown using dotted lines with wide spaces.

B. Regularities and Dependencies in the HP Model

We illustrate the emergence of regularities in the search space of the HP model using the $HHHPHPPPPPH$ sequence, introduced in [51]. For this sequence, and using the representation previously introduced, we find all possible solutions and evaluate them according to the HP and functional protein energy functions. The number of solutions evaluated are $3^9 = 19683$. Of these configurations, 8658 are not self-avoiding and they are assigned a very high energy equal to 100. In Table I, $H(\mathbf{x})$ denotes all the possible values that the two evaluated energy functions can reach for sequence $HHHPHPPPPPH$. HP_f and FP_f, respectively, indicate the number of solutions where the corresponding value of $H(\mathbf{x})$ has been achieved for the HP and functional protein energy functions.

It is important to highlight that there is not a one-to-one mapping between each solution and each state of the sequence. The reason is that one state can have more than one solution representation, i.e., the representation is redundant. For instance, while there exists only one optimal state for the functional model protein, in our representation, this optimal state has two possible representations corresponding to symmetrical configurations. In the case of the HP model, there are 16 optimal solutions. Fig. 3 shows optimal configurations for the functional model protein and the HP model.

To associate a probability value with every point of the search space, we will use a theoretical benchmark based on the Boltzmann distribution. In this benchmark, the probability of each solution is equal to the Boltzmann distribution calculated from the energy evaluation

$$p(\mathbf{x}) = \frac{e^{\frac{-H(\mathbf{x})}{t}}}{\sum_{\mathbf{x}'} e^{\frac{-H(\mathbf{x}')}{t}}} \tag{2}$$

where $H(\mathbf{x})$ is the energy of $\mathbf{x}$ and t is the temperature.

TABLE II
MARGINAL PROBABILITY DISTRIBUTIONS FOR (X_3, X_4, X_5) CALCULATED FROM THE BOLTZMANN DISTRIBUTIONS FOR HP AND THE FUNCTIONAL MODEL PROTEIN FOR SEQUENCE $HHHPHPPPPPH$

	$p_{HP}(x_3, x_4, x_5)$			$p_{FP}(x_3, x_4, x_5)$		
Seq	$0--$	$1--$	$2--$	$0--$	$1--$	$2--$
-00	**0.000**	0.057	<u>0.069</u>	**0.000**	0.091	<u>0.151</u>
-01	0.035	0.036	0.038	0.009	0.029	0.033
-02	0.035	0.036	0.035	0.009	0.029	0.026
-10	0.029	0.034	0.035	0.021	0.026	0.027
-11	0.038	0.040	0.038	0.033	0.034	0.033
-12	0.035	0.034	0.029	0.027	0.025	0.021
-20	0.035	0.036	0.035	0.026	0.026	0.009
-21	0.038	0.036	0.035	0.034	0.030	0.009
-22	<u>0.069</u>	0.057	**0.000**	<u>0.151</u>	0.091	**0.000**

Equation (2) shows the expression of the Boltzmann distribution. We set the temperature at 1, but t can be changed to simulate different experimental conditions. The Boltzmann distribution is a natural candidate for the fitness distribution. From a theoretical point-of-view, the Boltzmann distribution allows one to associate probabilistic independence properties between the variables of the problem with certain characteristics of the energy function [66]. It exhibits another convenient feature: higher probabilities are associated with points of the search space with better function evaluation. Therefore, these probabilities describe the desired performance of an ideal optimization algorithm: better points are visited with a higher probability. The Boltzmann distribution has also been used together with Monte Carlo-based methods for HP model optimization [55].

From the Boltzmann distribution, it is possible to compute the marginal probability distribution of any subset of variables by marginalization. This process is feasible in our case because the total number of configurations (3^9) is relatively small. Therefore, we calculate the marginal probability distributions corresponding to variables (X_3, X_4, X_5) for probability distributions p_{HP} and p_{FP}, which are shown in Table II. The table shows the marginal probabilities of the 3^3 configurations (from 000 to 222) of variables (X_3, X_4, X_5).

In Table II, the lowest probability values are bold-faced, while the highest values are underlined. The two configurations with zero probability[1] are shown in Fig. 4, where the direction of the sequence is represented as an arrow between the first and second residue of the sequence. Not surprisingly, these are the only two self-intersecting configurations that can be formed with three contiguous moves. Any solution that contains subchains 000 or 222 is not self-avoiding, and therefore receives a very low probability.

The two configurations with the highest probabilities are shown in Fig. 5. These are two symmetrical helices. In the HP and functional model protein, these type of substructures can give an important contribution to the final energy. Helices are present in the optimal solutions for both models, shown in Fig. 3.

A remarkable difference between the marginal probabilities corresponding to the HP and functional model protein is related to the probabilities given to the helices. The difference between the probabilities of the best and second best configurations is 0.012 for the HP model, and 0.06 for the functional model protein. This gives an idea of the difference due to the energy function used.

One conclusion from this experiment is that if we assume that subsolutions with highest probabilities in the model are those that represent optimal problem substructures (i.e., those likely to be present in the optimal solutions), then we can identify optimal substructures by inspecting the marginal distributions from the search distribution. These substructures are likely to be present in those population-based optimization algorithms able to respect the relevant interactions between the variables. In EDAs, the mapping between the problem structure and the probabilistic dependencies represented in the structure and parameters of the probabilistic models allows the use of the models as a source of information about the problem.

We investigate the effect that disregarding the potential interactions between variables may have on the modeling of the problem. In the next experiment, a univariate probability approximation of $p(x_3, x_4, x_5)$ is calculated. First, univariate marginal distributions are calculated for the three variables, $p(x_i) = \sum_{X_i = x_i} p(\mathbf{x})$. Afterwards, the approximation $p_a(x_3, x_4, x_5)$ is computed as the product of the univariate marginals $p_a(x_3, x_4, x_5) = p(x_3)p(x_4)p(x_5)$. The approximation is shown in Table III.

As in Table II, the lowest probability values are bold-faced, while the highest values are underlined. Due to the effect of rounding, some other configurations appear in the table with equal (rounded) probabilities than the lowest (respectively, highest) ones.

In Table III, it can be observed that the best and worst configurations do not agree with the ones obtained using the whole three-variate marginal probability distribution. The univariate approximation is not able to capture the structural features of the problem represented in Table II. This experiment illustrates the convenience of using higher order interactions to capture relevant features of the problem structure. As it has been analyzed in previous sections, traditional crossover operators do not respect these interactions. Furthermore, as explicit modeling of the search space is missing in most of nature-inspired algorithms, it is impossible to detect, represent, and store these regularities efficiently. In the following sections, we show how different probability models can detect and exploit this information.

The experiments presented in this section have been conducted using a single HP instance. Obviously, there are other factors that influence the marginal probability distributions corresponding to the different energy models. We have focused on showing the way in which structural regularities are exposed by the probability models learned. The analysis of other factors is beyond the scope of this paper.

[1]Strictly speaking, the probabilities are never zero but they approximate this value.

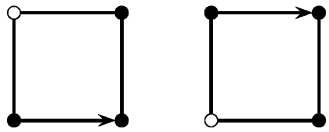


Fig. 4. Self-intersecting short paths: configurations with zero probability.

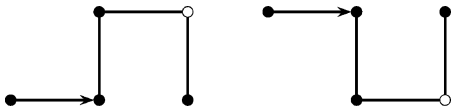


Fig. 5. Short helices: configurations with the highest probability.

TABLE III
UNIVARIATE APPROXIMATION OF THE MARGINAL PROBABILITY OF (X_3, X_4, X_5) CALCULATED FROM THE BOLTZMANN DISTRIBUTIONS FOR HP AND FUNCTIONAL MODEL PROTEIN FOR SEQUENCE $HHHPHPPPPPH$

	$p_{HP}(x_3,x_4,x_5)$			$p_{FP}(x_3,x_4,x_5)$		
Seq	$0--$	$1--$	$2--$	$0--$	$1--$	$2--$
-00	0.036	0.042	0.036	0.044	0.029	0.044
-01	0.036	0.042	0.036	0.029	**0.019**	0.029
-02	0.036	0.042	0.036	0.044	0.029	0.044
-10	**0.033**	0.038	**0.033**	0.054	0.035	0.054
-11	0.033	0.039	0.033	0.035	0.023	0.035
-12	**0.033**	0.038	**0.033**	0.054	0.035	0.054
-20	0.036	0.042	0.036	0.044	0.029	0.044
-21	0.036	0.042	0.036	0.029	**0.019**	0.029
-22	0.036	0.042	0.036	0.029	0.029	0.044

VI. EDAs FOR PROTEIN STRUCTURE PREDICTION

The existence of regularities in the search space, expressed in probabilistic dependencies between subsets of variables naturally leads to the convenience of using EDAs to take advantage of these regularities by capturing the dependencies. Probability models used by EDAs are built from the selected set of solutions. Therefore, the type of selection method used also influences the number and strength of the interactions learned by the model. Similarly to the Boltzmann distribution analyzed before, selection methods assign higher selection probabilities to solutions with higher fitness value.

In this section, we detail the main contributions of this paper: the introduction of EDAs to face the solution of the protein structure prediction problem, and the definition of the EDA-based model of protein folding. The section starts by introducing the probability models used by EDAs and explaining the rationale behind our choice. Finally, we define the EDA-based model of protein folding and describe the analogies between this model and some of the known behavioral characteristics of protein folding.

A. Probabilistic Models and Algorithms

We propose three types of probabilistic models to be applied to the protein structure prediction problem. In every case, solutions are represented using the vector representation introduced in Section V-A. Probabilistic models are presented together with the EDA that uses the model. EDAs are named according to the probability model that they use.

The first model considered is a k-order Markov model in which the configuration of variable X_i depends on the configuration of the previous k variables, where $k \geq 0$ is a parameter of the model. The joint probability distribution can be factorized as follows:

$$p_{MK}(\mathbf{x}) = p(x_1, \ldots, x_{k+1}) \prod_{i=k+2}^{n} p(x_i | x_{i-1}, \ldots, x_{i-k}). \quad (3)$$

The pseudocode of the Markov EDA $(\mathrm{MK-EDA}_k)$ is shown in Algorithm 2. The main step is the parametric learning of the probabilistic model. Since the structure of the Markov model is given, this step comprises to calculate the frequencies from the set of selected individuals and to compute the marginal and conditional probabilities. To sample a solution, first variables in the factor $(x_1, \ldots, x_{k+1})$ are generated and the rest of variables are sampled according to the order specified by the Markov factorization.

Algorithm 2: **Markov-EDA**

1. $D_0 \leftarrow$ Generate M individuals randomly
2. $l = 1$
3. **do** {
4. $D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to a selection method
5. Compute the marginal and conditional probabilities corresponding to each factor of factorization (3)
6. $D_l \leftarrow$ Sample M individuals (the new population) from the k-order Markov model
7. } **until** A stop criterion is met

The second probabilistic model is based on a tree where each variable may depend on no more than one variable that is called the parent. A probability distribution $p_{\mathrm{Tree}}(\mathbf{x})$ that is conformal with a tree is defined as

$$p_{\mathrm{Tree}}(\mathbf{x}) = \prod_{i=1}^{n} p\left(x_i | pa(x_i)\right) \quad (4)$$

where $Pa(X_i)$ is the parent of variable X_i in the tree, and $p(x_i|pa(x_i)) = p(x_i)$ when $Pa(X_i) = \emptyset$, i.e., when X_i is the root of the tree. The distribution $p_{\mathrm{Tree}}(\mathbf{x})$ itself will be called

a tree model when no confusion is possible. Probabilistic trees are represented by acyclic connected graphs.

The pseudocode of the tree-based EDA (Tree-EDA) is shown in Algorithm 3.

Algorithm 3: **Tree-EDA**

1. $D_0 \leftarrow$ Generate M individuals randomly
2. $l = 1$
3. **do** {
4. $D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to a selection method
5. Compute the univariate and bivariate marginal frequencies $p^s_i(x_i|D^s_{l-1})$ and $p^s_{i,j}(x_i, x_j|D^s_{l-1})$ of D^s_{l-1}
6. Calculate the matrix of mutual information using the univariate and bivariate marginals
7. Calculate the maximum weight spanning tree from the matrix of mutual information
8. Compute the parameters of the model
9. $D_l \leftarrow$ Sample M individuals (the new population) from the tree
10. } **until** A stop criterion is met

As presented in Algorithm 3, the bivariate probabilities are initially calculated for every pair of variables. From these bivariate probabilities, the mutual information between variables is found. To construct the tree structure, an algorithm introduced in [75], that calculates the maximum weight spanning tree from the matrix of mutual information between pairs of variables is used. To sample the solutions from the tree, we have used probabilistic logic sampling (PLS) [76], first the root variable is instantiated, and following the tree structure each variable is sampled conditioned on its parent.

The third model considered is a mixture of trees [77]. A mixture of trees is defined as a distribution of the form

$$p_{\mathrm{MT}}(\mathbf{x}) = \sum_{j=1}^{m} \lambda_j p^j_{\mathrm{Tree}}(\mathbf{x}) \tag{5}$$

with $\lambda_j > 0, j = 1, \ldots, m, \sum_{j=1}^{m} \lambda_j = 1$.

In this case, the tree distributions are the mixture components. The m trees may have different structures and different parameters.

$\mathrm{MT-EDA}_m$ is the name given to the EDA that uses a mixture of trees, m being the number of components in the mixture. $\mathrm{MT-EDA}_m$'s pseudocode is shown in Algorithm 4.

To learn a mixture of trees that gives a good approximation of the selected individuals we use an expectation-maximization (EM) mixtures of trees learning algorithm that was originally introduced in [77]. The idea underlying the EM algorithm is to compute and optimize the expected value of a likelihood function which is the log-likelihood of both, the observed and the unobserved data, given the current model estimate. The EM algorithm alternates the expectation and maximization steps until one stop condition is satisfied. To learn the structure of each component, the tree learning method used by Algorithm 3 is employed. More details about the mixture of trees learning algorithm can be found in [77] and [78]. To sample the solutions from each component the PLS method is used.

Algorithm 4: **Mixtures of Trees EDA**

1. $D_0 \leftarrow$ Generate M individuals randomly
2. $l = 1$
3. **do** {
4. $D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to a selection method
5. Compute a mixture of trees Q using the EM algorithm.
6. Compute the parameters of the model
7. $D_l \leftarrow$ Sample M individuals (the new population) from Q
8. } **until** A stop criterion is met

The probability models used by the previous EDAs can be separated into two classes according to the part of the problem structure that they exploit. The first class of probability models is based on the existence of connected neighbors. The assumption behind the use of Markov models is that the most important source of problem interactions comes from the connected neighbors. Markov models have been used in computational biology to identify coding regions in proteins, to align sequences, and to predict the protein secondary structure.

The second class of models allows for the existence of arbitrary connections between the variables of the problem subject to the representation constraints determined by the probabilistic models. This choice of the models tries to capture interactions arising from both, connected and topological neighbors. Therefore, algorithms that learn the structure of the model from the data [75], [77] are incorporated. Models that belong to this class differ in the type of structural constraints they represent.

Initial results of an EDA based on the Markov model, for the solution of 2-d lattice problems, were presented in [27]. These EDAs make a parametric learning of the model. An EDA called combining optimizers with mutual information trees, which searches for probabilistic models that can be represented using tree-shaped networks was introduced in [65]. MT-EDA was originally presented in [78]. All these algorithms have mainly been applied to binary problems. Poor results for preliminary experiments conducted using EDAs based on unconstrained Bayesian networks [72] determined to discard these algorithms from our experimental benchmark.

B. Implementation

In the chosen representation, there might be invalid vectors that correspond to self-intersecting sequences. To enforce the

validity of the solutions, we employ a variation of the backtracking method used in [21]. A solution is incrementally repaired in such a way that the self-avoidance constraint is fulfilled. At position i, the backtracking call is invoked only if self-avoidance cannot be fulfilled with any of the possible assignments to X_i. The order in which values are assigned to each variable is random. If all the possible values have been checked, and self-avoidance is not fulfilled yet, backtracking is invoked.

On the other hand, if the number of backtracking calls have reached a prespecified threshold, the repair procedure is abandoned. This is a compromise solution for situations in which the repair procedure can be too costly in terms of time. The threshold for the number of backtracking calls was set to 500 and this value was determined empirically. Further details about the original backtracking algorithm can be found in [21].

In our implementation of EDAs,[2] the truncation selection of parameter $T = 0.1$ is used. Let M be the population size. In this type of selection, the best $N = T \cdot M$ individuals, according to their function evaluations, are selected. We use best elitism, a replacement strategy where the population selected at generation t is incorporated into the population of generation $t + 1$. Thus, only $M - N$ individuals are generated at each generation except the first one. All the EDAs have been implemented in C++ language. All the experiments have been executed in a Pentium III processor with 933 MHz.

C. Computational Cost of the Algorithms

In this section, we analyze the computational cost of the different steps of the EDAs proposed for the protein structure prediction problem. The complexity of some of the steps are common to all the algorithms. Since the complexity of the algorithms depends on the cardinality of the variables, we use $r_{\mathrm{MAX}} = \max_{i \in 1,\ldots,n} |X_i|$ to represent the highest cardinality among the variables.

1) Initialization: The initialization step of all the algorithms consists in randomly initializing all the solutions in the first population. It has complexity $O(nM)$.

2) Evaluation: The computational cost of this step is problem dependent. It will depend on the function implementation. Let cost_f be the running time associated to the evaluation of function f, the running time complexity of this step is $O(M\mathrm{cost}_f)$.

3) Selection: The complexity of the selection step depends on the selection method used. Truncation selection consists on selecting the τM best individuals of the population. In the worst case, the complexity of this step is $O(M \log(M))$.

4) Probabilistic Model Learning Algorithms: The cost of the learning step changes for each algorithm.

- $\mathrm{MK} - \mathrm{EDA}_k$: Computing the marginal probabilities used by $\mathrm{MK} - \mathrm{EDA}_k$ implies computing the $n - k$ marginal probability tables from the selected population, it has complexity $O((n - k)N)$.
- Tree-EDA: Computing the bivariate marginal probabilities used by Tree-EDA has complexity $O(n^2 N)$. The calculation of the mutual information has complexity $O(n^2 r^2_{\mathrm{MAX}})$. The step that returns the maximum weight spanning tree has complexity $O(n^2)$. The total complexity of Tree-EDA is $O(n^2 N + n^2 r^2_{\mathrm{MAX}})$.
- $\mathrm{MT} - \mathrm{EDA}_m$: The running time of EM for mixtures of trees was calculated in [77]. The total time complexity is $O(mn^2 N + mn^2 r^2_{\mathrm{MAX}})$. Notice that, as expected, the complexity is equal to the complexity or learning a tree scaled by the number of components m.

5) Sampling: The cost of the sampling step changes for each algorithm. $\mathrm{MK} - \mathrm{EDA}_k$: Sampling M individuals from the Markov models has a complexity order $O((n - k)Mr^k_{\mathrm{MAX}})$.

Tree-EDA: Sampling M individuals with a tree has a complexity order $O(nMr^2_{\mathrm{MAX}})$.

$\mathrm{MT} - \mathrm{EDA}_m$: Sampling M individuals with a mixture of trees has a complexity order $O(mnMr^2_{\mathrm{MAX}})$.

6) Total Computational Costs of the Algorithms: The total computational costs of algorithms $\mathrm{MK} - \mathrm{EDA}_k$, Tree-EDA, and $\mathrm{MT} - \mathrm{EDA}_m$ are, respectively, $O(G((n - k)Mr^k_{\mathrm{MAX}} + M\mathrm{cost}_f))$, $O(G(n^2 M + n^2 r^2_{\mathrm{MAX}} + nMr^2_{\mathrm{MAX}} + M\mathrm{cost}_f))$, and $O(G(mn^2 M + mn^2 r^2_{\mathrm{MAX}} + mnMr^2_{\mathrm{MAX}} + M\mathrm{cost}_f))$, where the population size M and the number of generations G change according to the difficulty of the problem. While the cost of $\mathrm{MK} - \mathrm{EDA}_k$ scales linearly with the number of variables, the costs of Tree-EDA, and $\mathrm{MT} - \mathrm{EDA}_m$ have a quadratical scaling.

D. EDAs as a Model of the Protein Folding Mechanism

The existence of a number of analogies between the "new" view of the protein folding mechanism and the way EDAs behave motivate us to analyze EDAs as a model of protein folding. Basically, we highlight these coincidences, drawing parallels between both entities, and investigating to what extent each of the entities can provide answers to questions arisen in the other domain.

To explain our model of the protein folding process, we will use the same representation introduced in Section V-A. We will assume that all solutions are feasible (i.e., self-intersecting paths are repaired). At first, during the real folding process, a protein can only be in one state at each time t. However, in EDAs, at each time more than one configuration can be part of the population. To cover this gap, we will assign the main role in modeling to the EDA probabilistic model. This resembles the "new" view of protein folding, where proteins are seen as an ensemble of rapidly interconverting conformations. At each time, a probability $p(\mathbf{x})$ can be associated with every possible configuration $\mathbf{x}$ of the sequence. This probability is related to the energy of the configuration, usually using the Boltzmann distribution defined in (2).

Consider that a given EDA shall model the protein folding process. Each generation of the EDA will be considered a time step of the folding process. We will assume that the probability of the sequence to fold to a given conformation is equal to the probability given to the same configuration by the probabilistic model of the EDA constructed at generation t.

[2]The EDAs programs are available from the authors upon request.

Starting from this assumption, we advance the following statements.

- Both the "new" view and the EDA define a sampling of the space of configurations.
- The sampling processes pursue to sample the sequence configurations with a probability dependent of the quality of their respective energy function evaluations.
- The goal of the EDA and the protein folding process is achieved when $p(\mathbf{x}) = 1$, where $\mathbf{x}$ is the protein native state.
- Both entities tend to preserve local favorable conformational features through successive generations (time steps).

In principle, some of the features presented above are not exclusive attributes of EDAs. For instance, other population-based methods (e.g., GAs) can be used to sample the space of sequence configurations. The preservation of local favorable configurations, that may correspond to autonomous folding units in the real protein folding process, can also be accomplished to a certain extent by different evolutionary methods. However, the advantage of EDAs is that the probability model which they employ treats phenomena such as solution disruption and frustration, which may arise in the protein folding process, more effectively. Although in GAs a probability distribution of the solutions is implicitly used for the search, this probability distribution is explicitly learned and used in EDAs.

As explained in previous sections, traditional crossover operators tend to disrupt the construction of relevant subsolutions. The probabilistic model used by EDA matches to the statistical nature of the ensemble of conformations. This model is a condensed description of the selected population and, under suitable conditions, it also matches well with the EDA population at time $t + 1$. The main advantage of an EDA model of protein folding is that it can provide not only global statistical information, but also information about the local conformations in the ensemble. This information can be appropriately combined to avoid the disruption of relevant subsolutions.

Let us exemplify this with the frustration problem. In frustrated systems, there are contacts that are locally unfavorable but that exist in the optimal solution, or there are favorable contacts that first must be broken to reach the optimal solution [43]. Analogous to a frustrated system would be a population-based method that tries to optimize a frustrated function.

Let $f(\mathbf{x})$ be a function that can be decomposed into the sum of local functions defined on (possibly overlapping) subsets of its variables. $f(\mathbf{x})$ is said to exhibit frustration when the point $\mathbf{x}$ where its global optimum is reached does not maximize one or more of the local functions.

These types of functions are difficult to optimize because finding the optima of the local subfunctions does not guarantee that the global optimum will be found by the combination of these optima. The role of frustration as a source of hardness for evolutionary algorithms has been discussed in [42]. In [66], it is shown that taking into account the interactions that arise between the variables of the problem, EDAs can optimize frustrated functions. Although the capacity of EDAs to deal with frustration also depends on the probability model employed, we emphasize (in this respect) the suitability of using EDAs over GAs.

In the section of experiments, devoted to the simulation of the protein folding process using EDAs, we investigate some of the features exhibited by the EDA-model that mimic the behavior of the protein folding process. The issues considered are the following.

1) Whether there is a correlation between the successful rate of EDAs and the contact order of the protein models.
2) Whether there is a relationship between the generation convergence of EDAs for the HP model, and the contact order of the optimal solution.
3) Whether there are differences in the rate of formation of native contacts, and if these differences are associated with their contact separation.

In our simulations, we will employ some of the order parameters commonly used to investigate the protein folding process, but adapted to the simplified models. For the functional model protein, the contact order is calculated as the average sequence separation of the HH contacts in the corresponding solution. For example, the contact order of the configurations shown in Figs. 2 and 3 (left) are, respectively, 3 and 6.

VII. Experiments

In this section, we present experiments on the use of the EDAs introduced in this paper. The section is divided into three main parts. First, the problem instances used for the HP model, and the benchmark used for the functional model protein are presented. In the second part, we present results on the protein structure prediction problem in the 2-d and 3-d regular lattices. Finally, this section presents the results of the study through EDA simulations of some factors related to the protein folding process.

A. Problem Benchmark

The HP instances used in our experiments, and shown in Table IV, have previously been used in [13], [18], [19], [21], [27], [28], and [79]. The values shown in Table IV correspond to the best-known solutions $(H(x^*))$ for the 2-d regular lattice. It is important to highlight that most randomly generated amino acid sequences do not behave like natural proteins, because the latter are products of natural selection. Likewise, most randomly generated sequences of H and P residues in the HP model do not fold to a single conformation [3].

We have used the functional model protein for the experiments with the EDA-based model and for evaluating EDAs as optimization algorithms. The existence of a unique native state for the instances of this model is a desired attribute, particularly for the experiments done using the EDA-based model. For this type of experiment, we employed a set of 15 545 functional model proteins[3] that were optimally embedded on a 2-d square lattice [51]. For each instance, the benchmark provides the energy of the unique native state, together with the energy value and number of structures that are in the first excited state (best suboptimum). The length for each sequence is 23.

[3]http://www.cs.nott.ac.uk/~nxk/HP-PDB/2dfmp.html.

TABLE IV
HP INSTANCES USED IN THE EXPERIMENTS. THE SEARCH SPACE OF EACH INSTANCE IS 2^n, WHERE n IS THE SIZE OF THE INSTANCE

inst.	size	$H(\mathbf{x}^*)$	sequence
$s1$	20	-9	$HPHPPHHPHHPHPHHPPHPH$
$s2$	24	-9	$HHPPHPPHPPHPPHPPHPPHPPHH$
$s3$	25	-8	$PPHPPHHP^4HHP^4HHP^4HH$
$s4$	36	-14	$P^3HHPPHHP^5H^7PPHHP^4HHPPHPP$
$s5$	48	-23	$PPHPPHHPPHHP^5H^{10}P^6$
			$HHPPHHPPHPPH^5$
$s6$	50	-21	$HHPHPHPHPH^4PHP^3HP^3HP^4$
			$HP^3HP^3HPH^4\{PH\}^4H$
$s7$	60	-36	$PPH^3PH^8P^3H^{10}PHP^3$
			$H^{12}P^4H^6PHHPHP$
$s8$	64	-42	$H^{12}PHPH\{PPHH\}^2PPH\{PPHH\}^2$
			$PPH\{PPHH\}^2PPHPHPH^{12}$
$s9$	85	-53	$H^4P^4H^{12}P^6H^{12}P^3H^{12}P^3$
			$H^{12}P^3HP^2H^2P^2H^2P^2HPH$
$s10$	100	-48	$P^6HPH^2P^5H^3PH^5PH^2P^4H^2$
			$P^2H^2PH^5PH^{10}PH^2PH^7$
			$P^{11}H^7P^2HPH^3P^6HPH$
$s11$	100	-50	$P^3H^2P^2H^4P^2H^3PH^2PH^2PH^4$
			$P^8H^6P^2H^6P^9HPH^2PH^{11}P^2$
			$H^3PH^2PHP^2HPH^3P^6H^3$

TABLE V
FUNCTIONAL MODEL PROTEIN INSTANCES

inst.	$H(\mathbf{x})$	sequence
$sf1$	-20	$PHPPHPPHHHHPPHPPHPHPPHH$
$sf2$	-17	$PHPPHPPHHHHHPPPPHPPHPPH$
$sf3$	-16	$HPHPHPHHHPPHPPPHPHHPPHH$
$sf4$	-20	$HHHPHHHPPHHPPPHPHPPHHHH$
$sf5$	-17	$PHPPPPPPHPHHPHPHHHHPHPH$
$sf6$	-13	$HHPHPPHPPPPHPPPPHPPPHHH$
$sf7$	-26	$PHPHHPHHHHHHPPHHHPHHHHH$
$sf8$	-16	$HPHPPPHHHHPHPPPPHPHPHHH$
$sf9$	-15	$PHPHHPHHPHHPHPHPHPPPPPH$
$sf10$	-14	$HPHPHPPPPPHHPPPHPHPHPHH$
$sf11$	-15	$PHPPHHHPHPPHPHHPHPPPPPH$

TABLE VI
RESULTS OF EDAS FOR HP INSTANCES IN THE $2-d$ LATTICE. S: PERCENTAGE OF TIMES THE BEST SOLUTION HAS BEEN FOUND, $\bar{g}$: AVERAGE NUMBER OF GENERATIONS NEEDED TO FIND THE OPTIMUM

		MK-EDA$_2$			Tree-EDA			MT-EDA$_4$		
inst.	$H(\mathbf{x}^*)$	$H(\mathbf{x})$	S	$\bar{g}$	$H(\mathbf{x})$	S	$\bar{g}$	$H(\mathbf{x})$	S	$\bar{g}$
$s1$	-9	-9	100	3.34	-9	100	2.96	-9	100	3.18
$s2$	-9	-9	100	3.68	-9	100	3.80	-9	100	3.84
$s3$	-8	-8	100	4.14	-8	88	5.72	-8	90	5.24
$s4$	-14	-14	8	8.75	-14	4	13.50	-14	16	9.50
$s5$	-23	-23	14	19.57	-23	18	23.66	-23	4	21.00
$s6$	-21	-21	86	11.72	-21	98	12.67	-21	96	12.95
$s7$	-36	-35	10	55.06	-35	12	982.00	-35	18	121.22
$s8$	-42	-42	6	31.18	-41	10	50.70	-42	12	78.16
$s9$	-53	-52	4	218.00	-51	2	1355.00	-50	4	2161.05
$s10$	-48	-46	6	922.50	-46	16	1445.70	-47	2	707.00
$s11$	-50	-47	2	215.00	-47	12	1778.80	-48	2	845.00

For the optimization experiments using the functional model protein, we have selected a subset of 11 instances from the benchmark. The selected instances have been previously used as testbed of optimization algorithms in [17] and [23]. The set of instances selected and their minimal energy values are shown in Table V.

B. Results for the HP Model in the Two-Dimensional Lattice

The first experiment consists of finding the optimum of sequences shown in Table IV using EDAs with different probability models. The EDAs described in Section VI ($\mathrm{MK-EDA_2}$, Tree-EDA, and $\mathrm{MT-EDA_4}$) are used in our experiments. All the algorithms use a population size of 5000 individuals and a maximum of 5000 generations. The results of the experiments are shown in Table VI, where S is the percentage of times the best solution has been found in 50 experiments, and $\bar{g}$ is the average number of generations needed to find the optimum. Since we use best elitism (the whole selected population is passed to the next population), the average number of function evaluations needed to reach the optimum can be calculated as $M + (M - N)(\bar{g} - 1)$. Using truncation selection ($T = 0.1$), the average number of function evaluations is $5000 + 4000(\bar{g} - 1)$.

The first remarkable result is that all EDAs are able to find the optimum solution for sequences $s1 - s6$. All the algorithms find the second best solution for sequence $s7$, the best or second best for sequence $s8$, and very good solutions for the rest of the longer sequences. There are two facts that help to put these results in perspective: EDAs do not use local optimizers that could improve the results, and the parameters of the algorithms have not been tuned for every instance.

1) Deceptive Instances for EDAs: Instance $s7$ is deceptive for EDAs. We investigate in detail the performance of the EDAs for this instance. Detailed research on the dynamics of EDAs for deceptive problems throws light on the limitations of these methods and could contribute to their improvement.

The optimum of sequence $s7$ is -36. There are many suboptimal solutions with value -35. Fig. 6, lower left, shows the optimum solution that cannot be found by the EDAs. The rest of the solutions are the suboptimal ones ($H(\mathbf{x}) = -35$) found by

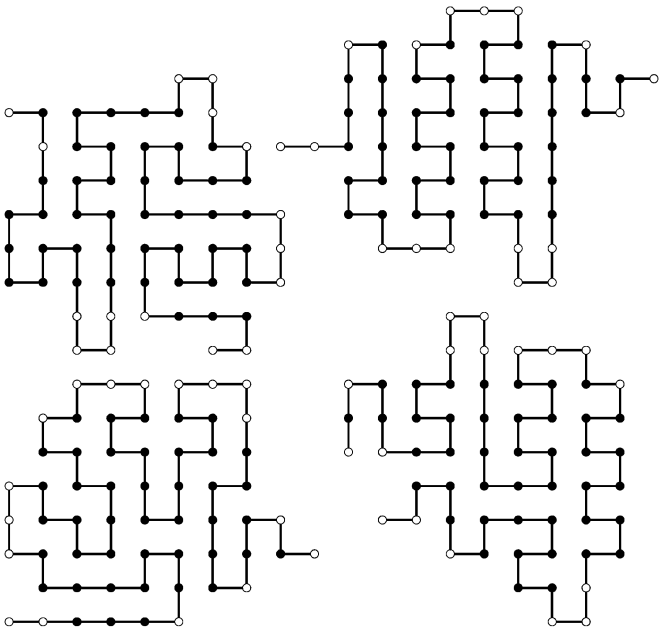


Fig. 6. Optimal solution (bottom left) and three suboptimal solutions for the $s7$ sequence.

the EDAs. A clear difference between the optimal and the rest of the solutions shown in the figure is the number of short helices. The optimum has fewer short helices than the other solutions. Most of the energy contributions come from interactions between a central cross-shaped structure and the neighboring residues. As the optimal solution cannot be constructed from the combination of the good substructures present in the other solutions, the EDA cannot reach it. As a hypothesis for the reason of this deceptive behavior, we advance the existence of isolated optimal solutions with components that are not present in the suboptimal solutions.

To validate this empirical conclusion, we calculate different Markov probabilistic models ($k = 0, \ldots, 3$) from the 5375 solutions with energy between -33 and -35. Using these models, the probabilities corresponding to each of the data base solutions and to the optimal solution shown in Fig. 6 are calculated. The models indicate the probability for the solutions to be present at the new EDA generation.

Results can be appreciated in Table VII. In this table, $p(\mathbf{x}_{\rm opt})$, $\max(p(\mathbf{x}))$, $\mathrm{mean}(p(\mathbf{x}))$, and $\min(p(\mathbf{x}))$, respectively, correspond to the probabilities given by the models with different k values (from 0 to 3) to the optimum of the problem, the maximum, mean, and minimum probabilities given by the models to the 5375 solutions. Similarly, $N(p(\mathbf{x}) > p(\mathbf{x}_{\rm opt}))$ is the number of solutions with a probability higher than the probability of the optimum.

The most revealing fact is that the probability given to the optimal solution by probability models with $k = 2$ and $k = 3$ is zero. This fact means that the optimal structure does not share some of its substructures with any of the other 5375 suboptimal solutions. The analysis of the model revealed that in the case of $k = 2$, only one of the substructures was absent in the other solutions, while when $k = 3$, four substructures were absent.

TABLE VII
STATISTICAL INFORMATION EXTRACTED FROM k-ORDER MARKOV PROBABILISTIC MODELS ($0 \leq k \leq 3$) OF THE 5375 SOLUTIONS OF THE $s7$ SEQUENCE WITH ENERGY LOWER THAN -32

k	$p(\mathbf{x}_{opt})$	$max(p(\mathbf{x}))$	$mean(p(\mathbf{x}))$	$min(p(\mathbf{x}))$	$N(p(\mathbf{x}) > p(\mathbf{x}_{opt}))$
0	$9.6546e^{-27}$	$2.0721e^{-20}$	$1.2853e^{-22}$	$2.9364e^{-36}$	952
1	$6.8792e^{-23}$	$2.1155e^{-11}$	$9.8979e^{-14}$	$9.0532e^{-37}$	4952
2	0	$1.7751e^{-06}$	$4.0864e^{-09}$	$1.9233e^{-30}$	5375
3	0	$1.4889e^{-05}$	$4.4991e^{-08}$	$1.9347e^{-27}$	5375

TABLE VIII
RESULTS ACHIEVED BY DIFFERENT SEARCH HEURISTICS FOR THE HP INSTANCES

	$BestEDA$	GA	MMA	ACO	$NewACO$	$PERM$
$inst.$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$
$s1$	-9	-9	-9	-9	-9	-9
$s2$	-9	-9	-9	-9	-9	-9
$s3$	-8	-8	-8	-8	-8	-8
$s4$	-14	-14	-14	-14	-14	-14
$s5$	-23	-22	-22	-23	-23	-23
$s6$	-21	-21		-21	-21	-21
$s7$	-35	-34		-34	-36	-36
$s8$	-42	-37		-32	-42	-38
$s9$	-52				-51	-53
$s10$	-47				-47	-48
$s11$	-48				-47	-50

This experiment shows that deception also arises in the case of the HP model, and that deceptive instances for the EDAs can be found and described as those which do not share a number of good substructures with most of the closest suboptimal solutions. These substructures cannot be captured by the probability models used.

2) Comparison With Other Algorithms: The performance of EDAs is compared now with the best results achieved with other evolutionary and Monte Carlo optimization algorithms. The results are shown in Table VIII. The results of GA [19] and MMA [17] correspond to the best solution found in five runs. The results of ACO and NewACO [28] are based on 20–700 tries for the former algorithm and 300–500 for the latter [28]. PERM [25] reports 20–200 tries for sequences $s9$ and $s11$; the other instances have been faced in [28]. Results for other optimization methods [15], [22], [50] were not displayed because either they were unable to find the best results achieved by EDAs or the number of functions evaluations required to find them was

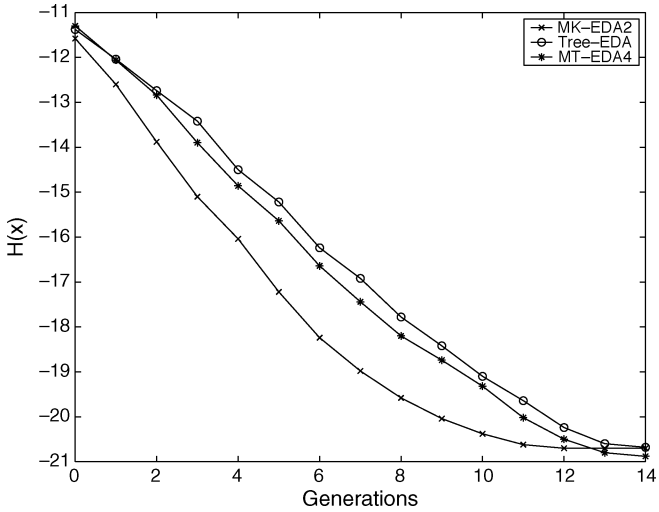


Fig. 7. Average of the best solution at each generation for different EDAs in sequence $s6$.

much higher. All the results shown for EDAs are obtained from 50 runs.

A first conclusion is that none of the algorithms are able to outperform the rest of algorithms for all the instances. PERM is one of the best contenders in all cases except $s8$ in which its results are consistently very poor. In comparison with the NewACO, EDA reaches equal or better results in all instances except one. It should be noted that NewACO applies local optimization techniques. In this sense, a fairer comparison would be between EDAs and ACO. In such a comparison, EDA is the clear winner. Analysis shows that none of the other algorithms achieves similar results.

3) Dynamics of the Algorithms: Another relevant issue related to EDAs concerns their particular dynamics for the HP models. The existence of a model of the search space enables a compact representation of characteristic features of the best solutions but also determines the particular way in which the optimal solutions are constructed.

In the next experiment, we analyze the convergence dynamic of the three types of EDAs employed in our experiments, in the optimization of sequence $s6$. The average of the best solutions at each generation is calculated by running each algorithm 50 times. The population size and the rest of the parameters were the same as in previous experiments. The results are shown in Fig. 7. It can be appreciated from the figure that, for all EDAs, a small number of generations are enough to find the optimal solutions. Nevertheless, there are differences in the dynamics of the algorithms. $\mathrm{MK}-\mathrm{EDA}_2$ reaches better solutions earlier than the other algorithms, but in the experiments conducted, $\mathrm{MT}-\mathrm{EDA}_4$ reached the optimum more often. Tree-EDA is the slowest algorithm.

C. Results of the HP Model in the Three-Dimensional Lattice

In the following experiments, we investigate the behavior of EDAs in the solution of the HP model in the regular 3-d lattice. EDAs are compared with the hybrid GA that uses a feasible-space approach with relative encoding [21] and with results achieved using the IA [23]. A maximum number of 10^5 evaluations was set in the experiments presented in [21] and [23]. Therefore, we imposed the same restriction and in our comparison EDAs with two possible instantiations of the parameters were used: A population size $N = 2500$ with $g = 40$ generations, and a population size $N = 5000$ with $g = 20$ generations. Since we use elitism with parameter $E = 1$ (i.e., the best individual is passed to the next generation), the maximum number of evaluations is slightly smaller than 10^5. The results of the experiments are shown in Table IX, where average and fitness values are computed from 50 runs for each algorithm. For each instance, the first row shows the results of EDAs with $N = 2500$ and the second those for $N = 5000$. Best and average results of the EDAs are in bold whenever they are equal or better than those achieved by the hybrid GA and the IA.

It can be seen in Table IX that in terms of the average fitness of the solutions, the results reached by all EDAs are strictly better than those achieved with the other two algorithms for instances s1, s2, s3, and s6. For instances s5, s7, and s8, the IA clearly outperforms all the EDAs in terms of the average fitness. For instance s4, the Tree-EDA and the $\mathrm{MT}-\mathrm{EDA}_4$ that use a population size of 2500 outperform the IA. For instances s6 and s8, EDAs are able to achieve best new solutions.

We hypothesized that the best solutions found by the algorithms for some of the instances may be far from optimal. EDAs that use a constrained population size cannot reach these optimal solutions. To validate this hypothesis, and in order to evaluate the capacity of EDAs to reach better solutions when more evaluations are allowed, we used a population size of 5000 individuals and a maximum of 1000 generations. In this case, we employ the best elitism strategy. We also run experiments for the hybrid GA allowing a maximum of $5 \cdot 10^6$ evaluations. The IA program was not available for conducting new experiments. The results are shown in Table X.

Since in this case we could collect the best result achieved in each run for all the algorithms, we could determine whether differences between the algorithms are statistically significant. We have used the Kruskal–Wallis test to accept or reject the null hypothesis that the samples have been taken from equal populations. The test significance level was 0.01. For all the instances considered, significant statistical differences have been found between the hybrid GA results and those achieved by the Markov $(\mathrm{MK}-\mathrm{EDA}_2)$ and mixture $(\mathrm{MT}-\mathrm{EDA}_4)$ EDAs. Significant statistical differences between the hybrid GA and the Tree-EDA have been detected only for instances $s3$, $s5$, $s6$, and $s8$. All the EDAs have a better average of solutions, showing that the algorithm clearly outperforms the hybrid GA. Furthermore, as can be observed in Table X, EDAs find new best solutions for sequences $s5$, $s6$, and $s8$. Notice that the average results for instances $s1$ and $s2$ degrade with regard to those shown in Table IX. We consider this fact may be due to the effect of the best elitism strategy that for small instances may lead to an early convergence of the algorithm. The average number of evaluations needed by EDAs before convergence was between $6 \cdot 10^4$ for the shortest instances and 10^6 for the longest ones.

D. Results for the Functional Model Protein

We have conducted experiments with the functional model protein in the two-dimensional lattice.

TABLE IX
RESULTS OF THE EDAS, THE HYBRID GA, AND THE IA IN THE THREE-DIMENSIONAL LATTICE

	hybrid GA		IA		MK-EDA$_2$		TreeEDA		MT-EDA$_4$	
	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$
$s1$	-11	-9.84 ± 0.86	-11	-10.90 ± 0.32	$\mathbf{-11}$	$\mathbf{-11.00 \pm 0.00}$	$\mathbf{-11}$	$\mathbf{-11.00 \pm 0.00}$	$\mathbf{-11}$	$\mathbf{-11.00 \pm 0.00}$
					$\mathbf{-11}$	$\mathbf{-11.00 \pm 0.00}$	$\mathbf{-11}$	$\mathbf{-10.96 \pm 0.04}$	-11	$\mathbf{-11.00 \pm 0.00}$
$s2$	-11	-10.00 ± 0.87	-13	-12.22 ± 0.65	$\mathbf{-13}$	$\mathbf{-12.94 \pm 0.09}$	$\mathbf{-13}$	$\mathbf{-12.86 \pm 0.16}$	$\mathbf{-13}$	$\mathbf{-12.70 \pm 0.50}$
					$\mathbf{-13}$	$\mathbf{-13.00 \pm 0.00}$	$\mathbf{-13}$	$\mathbf{-12.96 \pm 0.04}$	$\mathbf{-13}$	$\mathbf{-13.00 \pm 0.00}$
$s3$	-9	-8.64 ± 0.69	-9	-8.88 ± 0.48	$\mathbf{-9}$	$\mathbf{-8.94 \pm 0.06}$	$\mathbf{-9}$	$\mathbf{-8.90 \pm 0.09}$	$\mathbf{-9}$	$\mathbf{-8.98 \pm 0.02}$
					$\mathbf{-9}$	$\mathbf{-8.96 \pm 0.04}$	$\mathbf{-9}$	$\mathbf{-8.98 \pm 0.02}$	$\mathbf{-9}$	$\mathbf{-8.98 \pm 0.02}$
$s4$	-18	-13.72 ± 1.41	-18	-16.08 ± 1.02	$\mathbf{-18}$	-15.66 ± 1.54	$\mathbf{-18}$	$\mathbf{-16.34 \pm 0.51}$	$\mathbf{-18}$	$\mathbf{-16.32 \pm 0.10}$
					$\mathbf{-18}$	-15.48 ± 0.83	-17	-15.00 ± 0.86	$\mathbf{-18}$	-15.02 ± 0.88
$s5$	-28	-18.90 ± 2.08	-28	-24.82 ± 0.71	-22	-19.66 ± 1.37	-27	-23.62 ± 1.83	-23	-18.44 ± 1.60
					-24	-20.52 ± 1.15	-24	-20.68 ± 1.65	-24	-20.22 ± 2.30
$s6$	-22	-19.06 ± 1.46	-23	-22.08 ± 1.43	$\mathbf{-30}$	$\mathbf{-26.30 \pm 2.26}$	$\mathbf{-30}$	$\mathbf{-26.00 \pm 2.82}$	$\mathbf{-28}$	$\mathbf{-26.70 \pm 1.97}$
					$\mathbf{-26}$	$\mathbf{-23.38 \pm 1.30}$	$\mathbf{-26}$	$\mathbf{-22.08 \pm 2.48}$	$\mathbf{-24}$	$\mathbf{-22.54 \pm 1.27}$
$s7$	-38	-32.28 ± 3.09	-41	-39.02 ± 0.50	-37	-32.66 ± 3.13	-37	-32.94 ± 1.53	-35	-31.72 ± 2.98
					-38	-33.84 ± 2.91	-38	-33.10 ± 3.11	-38	-32.46 ± 3.03
$s8$	-36	-30.84 ± 2.55	-42	-39.07 ± 1.20	$\mathbf{-42}$	-36.66 ± 4.02	$\mathbf{-44}$	-34.70 ± 6.87	-37	-32.24 ± 2.47
					-40	-34.66 ± 2.60	-34	-30.82 ± 2.97	-37	-30.96 ± 2.47

TABLE X
RESULTS OF THE EDAS AND THE HYBRID GA IN THE THREE-DIMENSIONAL LATTICE

	hybrid GA		MK-EDA$_2$		Tree-EDA		MT-EDA$_4$	
	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$	$H(\mathbf{x})$	$mean \pm \sigma$
$s1$	-11	-10.52 ± 0.54	-11	-10.82 ± 0.38	$\mathbf{-11}$	-10.68 ± 0.51	$\mathbf{-11}$	$\mathbf{-10.84 \pm 0.37}$
$s2$	-13	-11.28 ± 0.90	$\mathbf{-13}$	$\mathbf{-12.02 \pm 0.94}$	$\mathbf{-13}$	-11.30 ± 0.85	$\mathbf{-13}$	-11.88 ± 0.93
$s3$	-9	-8.54 ± 0.64	$\mathbf{-9}$	-8.96 ± 0.19	$\mathbf{-9}$	-8.92 ± 0.27	$\mathbf{-9}$	$\mathbf{-9.00 \pm 0.00}$
$s4$	-18	-15.76 ± 1.05	$\mathbf{-18}$	-16.40 ± 0.80	$\mathbf{-18}$	-16.24 ± 0.83	$\mathbf{-18}$	$\mathbf{-16.50 \pm 0.96}$
$s5$	-28	-24.60 ± 1.57	$\mathbf{-29}$	$\mathbf{-27.24 \pm 0.92}$	$\mathbf{-29}$	-26.88 ± 0.93	$\mathbf{-29}$	-27.06 ± 1.08
$s6$	-26	-23.02 ± 1.48	-29	-25.70 ± 1.26	$\mathbf{-31}$	$\mathbf{-25.94 \pm 1.58}$	-28	-25.74 ± 1.22
$s7$	-49	-41.18 ± 2.75	$\mathbf{-49}$	$\mathbf{-46.30 \pm 2.04}$	$\mathbf{-49}$	-43.78 ± 3.10	-48	-42.00 ± 6.76
$s8$	-46	-40.40 ± 2.50	$\mathbf{-52}$	$\mathbf{-46.78 \pm 2.28}$	-49	-43.72 ± 2.43	-50	-45.64 ± 2.03

Initial results for the functional model protein instances [17] considered the number of fitness evaluations required by the best run for a given instance as a metric for comparing the performance of the algorithms. In [23], the success rate and the number of fitness evaluations are used to compare different variants of the IA. We follow the second approach and compare the results of EDAs with three of the variants of the IA presented in [23]. The variants of the IA selected were: IA with elitist aging, IA with pure aging, and IA using memory cells ($\tau_B = 5$, $\tau_{Bmem} = 10$). The other two variants for which results are shown in [23] do not improve the results of the IA variants selected.

For EDAs, we use a scheme conceived for situations wh2ere early convergence of the algorithm is detected. This scheme works as follows: When the selected population is too homogenous (the number of different individuals is below a given threshold), all the individuals except the best solution are randomly generated. This scheme allows one the use of smaller population sizes. We use a population size $N = 500$, and the minimal number of different individuals allowed was 50. As in [23], a maximum of $5 \cdot 10^6$ evaluations were allowed to all the algorithms and 30 runs were done for each instance. Results are shown in Table XI.

In Table XI are shown, for all the algorithms, the success rate percentage and the average number of evaluations, which are the criteria we use to compare the algorithms. An initial analysis of the results shown in the table reveals that IA with elitism aging and IA with memory cells have the worst results among

TABLE XI
RESULTS OF THE EDAS AND THE IA FOR THE FUNCTIONAL MODEL PROTEIN IN THE TWO-DIMENSIONAL LATTICE. S: PERCENTAGE OF TIMES THE BEST SOLUTION HAS BEEN FOUND, $\bar{g}$: AVERAGE NUMBER OF FITNESS EVALUATIONS NEEDED TO FIND THE OPTIMUM

	IA Elitism Aging		IA Pure Aging		IA Memory Cell		MK-EDA$_2$		Tree-EDA		MT-EDA$_4$	
inst.	S	$\bar{f}$	S	$\bar{f}$	S	$\bar{f}$	S	$\bar{f}$	S	$\bar{f}$	S	$\bar{f}$
$sf1$	100	45563	100	32647.73	100	38586.63	100	**25799.3**	100	29841.2	100	86111.8
$sf2$	100	40389.47	100	**17526.73**	100	28434.9	100	20593.1	100	22023.5	100	50699.4
$sf3$	3.33	1194332	56.67	2403985.3	43.33	2583300.8	100	**140802**	100	997618	100	835909
$sf4$	100	168227.83	100	128015.1	100	130849	100	**108118**	100	371789	100	795390
$sf5$	100	17667.9	100	**12095.33**	100	20834.46	100	242947	100	55772.6	100	197156
$sf6$	83.33	2732830	100	332938.5	100	483126.76	100	197156	100	**61577.6**	100	197156
$sf7$	40	127829.83	100	**584179.8**	46.67	588057.5	86.67	3566200	63.33	4277590	80	3433060
$sf8$	100	149415.4	100	**38262.6**	100	42562.53	100	1373380	100	147938	100	901361
$sf9$	10	377108	100	281720.8	93.33	907962.4	70	3418170	100	2136650	100	**155722**
$sf10$	100	587811	100	104155.4	100	100085.43	100	518495	100	**57652.1**	100	159415
$sf11$	93.33	174583.82	100	27743.7	100	71903.1	100	14621.7	100	**11927.1**	100	37708.8

all the algorithms tested. However, there is not a clear winner among the rest of algorithms. $\mathrm{MT-EDA}_4$ has the highest success rate of all the algorithms but for many instances it needs a higher number of function evaluations than IA with pure aging, which is the best of all the IA variants. Differences between EDAs are less evident. The $\mathrm{MT-EDA}_4$ and Tree-EDA only fail for one instance and the $\mathrm{MK-EDA}_2$ for two. Some of the results shown in Table XI have been improved by using the $\mathrm{MK-EDA}_1$ (data not shown). The results achieved for the functional model protein in the two-dimensional lattice show that for this type of problem, EDAs are very competitive algorithms.

E. Results of the Protein Folding Simulations

In this section, we present results on the use of EDAs to simulate the protein folding in the HP model. We investigate to which extent EDAs are able to mimic some characteristic features of the protein folding mechanism. For all the simulation experiments, we use $\mathrm{MK-EDA}_2$. In all the experiments, the population size was set at 2000 and the maximum number of generations was 20.

1) Energy Landscape of the Model: In the first experiment, we investigate the energy landscape of the function for one of the 15 545 functional model protein instances available. The goal of this experiment is to build some intuition about the fitness landscape of the functional model protein. Particularly, to illustrate the fact that as solutions share more of the protein native contacts, their fitness (energy) decreases.

The experiments consist of executing the EDA and storing all the conformations visited during the search. Sequence $PHHPPHPPHHPPHPHPPHPPHHH$ was chosen for the experiment. For every conformation visited during the search, the total number of contacts (K) and the number of native contacts (Q) are calculated. We classify the conformations using these two parameters, and calculate the average energy of all conformations that share the parameters (K, Q). The native state of the sequence has nine native contacts. Therefore, $0 \leq K, Q \leq 9$.

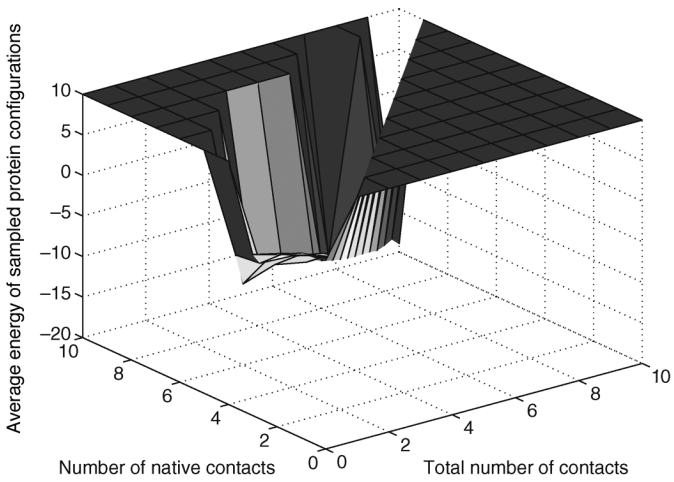


Fig. 8. Energy landscape of the functional model protein corresponding to sequence $PHHPPHPPHHPPHPHPPHPPHHH$ as sampled by $\mathrm{MK-EDA}_2$.

Fig. 8 shows the average energy of all the sampled conformations grouped using the different values of K and Q. The figure reveals that, as the number of native contacts increases in the conformations, the average energy decreases.

Throughout the evolution, $\mathrm{MK-EDA}_2$ is able to explore the different regions of the energy landscape. Not only those regions with high energy which are abundant in the search space, but also those corresponding to low-energy values where the number of conformations is scarce. For example, $\mathrm{MK-EDA}_2$ is able to locate the optimum which is unique.

To appreciate the characteristics of this landscape in detail, Fig. 9 shows the contour graph corresponding to the same experiment. Sets of conformations with similar average energy are

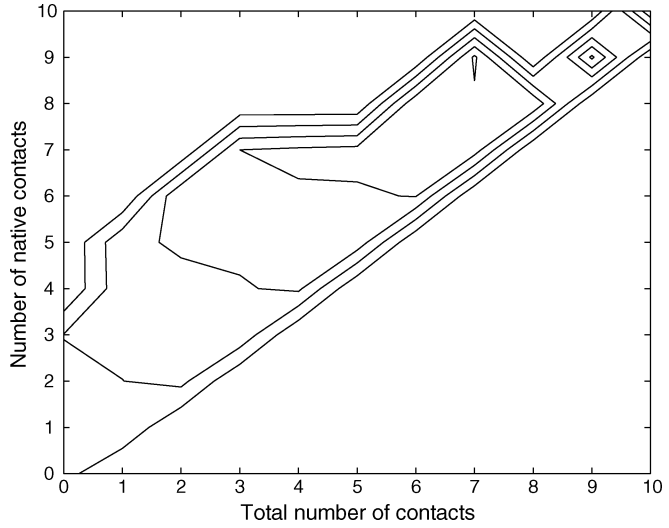


Fig. 9. Energy landscape contour graph of the functional model protein corresponding to sequence $PHHPPHPPHHPPHPHPPHPPHHH$ as sampled by $\mathrm{MK-EDA}_2$.

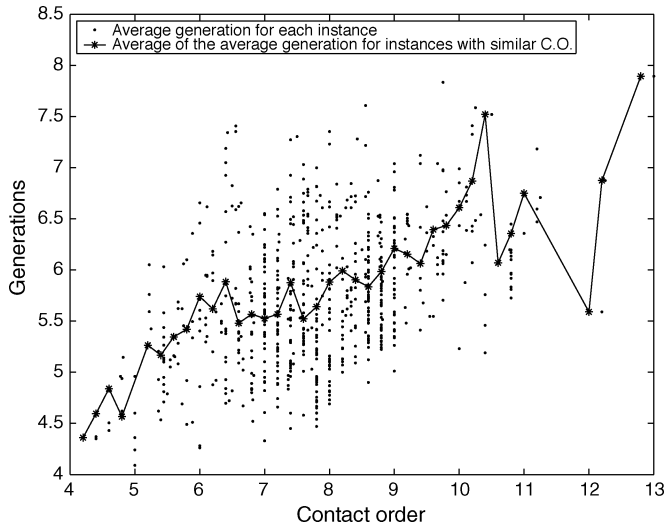


Fig. 10. Relationship between the contact order of the sequences where $\mathrm{MK-EDA}_2$ has a success rate above 95 in 100 experiments and the average number of generations needed to convergence.

joined in the graph by the same contour lines. Regions with lower energy are those with many native contacts.

The samples obtained by $\mathrm{MK-EDA}_2$ make it possible to observe the correlation between the presence of native contacts in the conformations and the quality of the folding (low energy). This fact is expected in proteins that have evolved to diminish the degree of frustration and achieve a fast folding rate. However, it does not mean that, for every simplified energy function, the EDA will be able to move in the direction of the native configuration by augmenting the number of native contacts. First, the definition of realistic protein folding energy functions such that their only significant basin of attraction is the native state has been recognized as a very difficult task [38]. Second, the phenomenon of frustration can arise in the functional model protein and other energy functions, moving the $\mathrm{MK-EDA}_2$ away from the native state.

2) Influence of the Contact Order in the Protein Folding Process: We investigate whether the EDA can reflect the influence of the contact order parameter in the protein folding process. As discussed in Section II-A, proteins with a large fraction of their contacts between residues close in sequence tend to fold faster than proteins with more nonlocal interactions. This section shows how the contact order of the functional model protein instances influences on the success rate and average number of generations needed by $\mathrm{MK-EDA}_2$ to solve the problem.

As a preprocessing step, 100 executions of $\mathrm{MK-EDA}_2$ are done for each of the 15 545 functional model proteins instances. The goal of this step is to determine the easier instances for the EDA, and to calculate, for all instances, a set of statistics from the best solutions found at each run. Of the 15 545 instances, $\mathrm{MK-EDA}_2$ was able to find the optimum in 12 588 instances at least once. For 703 instances, the algorithm found the optimum at least in 95 runs, and for 176 it found the optimum in 100 runs.

To evaluate the relationship between the contact order of the sequences and the average number of generations needed to solve them, we consider those instances for which $\mathrm{MK-EDA}_2$ has a success rate higher than or equal to 95. The choice of this set is determined by the need to have an accurate estimate of the average number of generations. From this set of instances, the Spearman's rank correlation coefficient is calculated between the contact order and the average number of generations. The Spearman's rank correlation is the statistic of a nonparametric test usually employed to test the direction and strength of the relationship between two variables.

With a confidence level of 0.01, the test accepted the hypothesis that the two measures have a positive correlation equal to 0.3939. In Fig. 10, the average generation and contact order for the set of selected instances are plotted. Additionally, the figure shows the average number of generations calculated from instances with similar contact order. It can be seen in the figure that the number of generations needed to solve the problem grows with the contact order.

Using the test based on the Spearman's rank correlation coefficient with the same confidence level, we have evaluated the relationship between the contact order of the sequences where $\mathrm{MK-EDA}_2$ found the optimum at least once, and the success rate achieved by the algorithm for these instances. The test rejected the hypothesis that the observed correlation was due to random effects. The parameter of the correlation provided by the test was equal to -0.2848. This means that, as the contact order of the instance grows, the success rate of the algorithm diminishes. Fig. 11 shows the average success rate for instances with similar contact order.

The analysis of the EDA-based protein folding model has shown that similarly to the real protein folding process, low contact order optimal conformations are easier and faster to find. This behavior has been observed in other optimization algorithms such as Rosetta [48], which is one of the best algorithms for protein folding in the CASP competitions. In Rosetta, the structures sampled by local sequences are approximated by the distribution of structures seen for short sequences and related

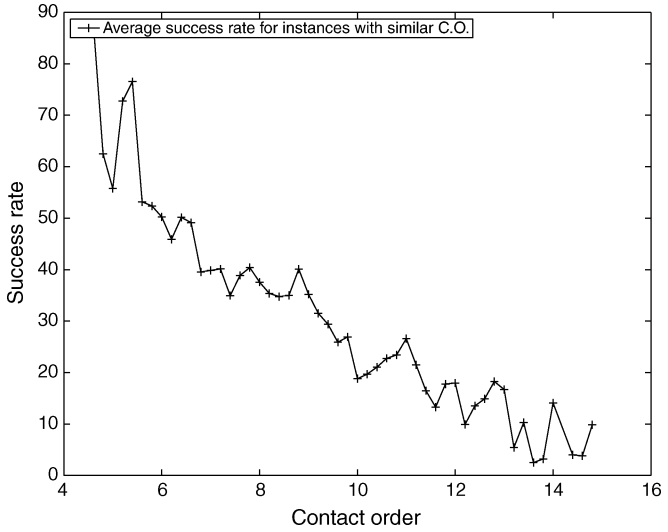


Fig. 11. Relationship between the contact order of the sequences where $\mathrm{MK}-\mathrm{EDA}_2$ found the optimum at least once, and the success rate achieved by the algorithm for these instances.

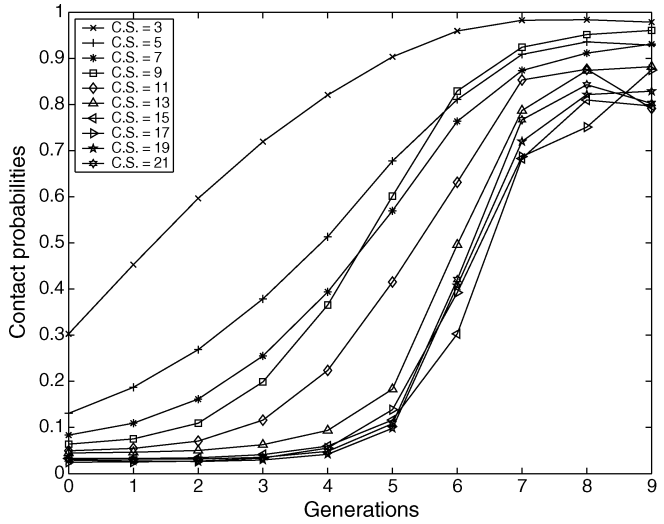


Fig. 12. Relationship between the different contact separations and the evolution of their probabilities along the evolution of $\mathrm{MK}-\mathrm{EDA}_2$ for instances where it had a 100% of success rate.

sequences in known protein structures. Using Rosetta, high contact order proteins can take up to six orders of magnitude longer to fold than do low contact order proteins [80].

3) Difference in the Contact Accessibility: In the following experiment, we investigate whether the EDA-based protein folding model is able to reproduce the difference in the rate of formation for different native contacts. Particularly, we will investigate whether the distance between contacting residues has an effect on their formation. As discussed in Section II-A, local interactions are more likely to form earlier in the real folding process than nonlocal interactions.

Given a protein instance that is optimized using $\mathrm{MK}-\mathrm{EDA}_2$, the experiment consists of computing, at each generation, the frequency of each native contact of the protein from the selected set of the EDA population. The frequency is calculated as the fraction of all selected solutions that contain that contact. The EDA is executed 100 times for each of the 176 instances for which $\mathrm{MK}-\mathrm{EDA}_2$ had previously found a 100% of success.

From the information obtained from the 17 600 experiments, the frequencies of the contacts with the same contact separation (C.S.) at the same generation are averaged. The results are shown in Fig. 12. By selecting a set of instances, we intend to show that the observed effect are not particular of one single instance. On the other hand, by choosing the 176 instances for which $\mathrm{MK}-\mathrm{EDA}_2$ had previously found a 100% of success rate, it is guaranteed that the native state will be reached in all the runs with a very high probability.

Fig. 12 shows the evolution of the probabilities along the generations. Two aspects can be appreciated. First, the probability of contacts with low contact separation $(C.S. \leq 5)$ is higher than the other probabilities since the first set of individuals is selected. Second, the difference in the rate of convergence determined by the contact separation is evident. While contacts with low contact separation rapidly increase their probability, those contacts with a higher contact separation $(C.S. \geq 15)$ grow at a slower rate.

The behavior of our model is once again consistent with what is observed in the real protein folding. Moreover, these results can support hypotheses that help to explain the correlation between the contact order of proteins and the success rate and average number of generations needed to find them. As the increase of the probability of contacts with high contact separation is slower, it will take a longer time to obtain native states with higher contact order. Additionally, it will also be more difficult to find the optimum for this type of instances as the algorithm will tend to get trapped in suboptimal solutions with lower contact order.

VIII. Conclusion and Further Work

The approach of using probabilistic dependencies to improve search efficiency has a strong theoretical basis. Its operational simplicity and applicability make it an advantageous method in relation to other widely applied evolutionary algorithms. The results of the experiments shown in this paper confirm that the EDA is a feasible alternative for the protein structure prediction problem. Particularly, we recommend the use of probabilistic models for the solution of coarse-grained protein folding problems, where Monte Carlo methods exhibit a poor performance.

There exist evolutionary and deterministic algorithms that exhibit similar or better results than EDAs for some of the HP and functional protein instances used in this paper. Some of these algorithms have been tuned for each of the instances or they implement local optimizers together with the main evolutionary method used. This is certainly an alternative that could be followed for EDAs too and might provide better results than those presented in the paper. Nevertheless, we have put the emphasis in the investigation of those factors that explain the behavior achieved by EDAs, and in which way problem knowledge can be obtained by inspecting the solutions found by the algorithms. We investigate the influence that different factors of the HP instances have on the behavior of the algorithm. The decision of not including other optimizers was also due to the interest to

simplify the analysis of the role played by probabilistic modeling in the search.

On the other hand, in comparison to other optimization algorithms such as PERM, which has only been applied to the HP-model, EDAs are general optimization strategies that have been extended to solve protein problems with more complex protein models [61], [62], [81]. Nevertheless, this work and other recent successful applications of EDAs in bioinformatics [27], [82]–[85] refer to the use of EDAs as optimization methods. One of the aims of this paper has been to highlight that EDAs can also be employed as simulation tools of biological processes.

The most distinguished feature of the EDA model of protein folding lies in its ability to represent, by means of probability models, relevant interactions between the protein conformations. This representation can be used in broader contexts. For instance, in the Rosetta algorithm, short fragments of known proteins are assembled by a Monte Carlo strategy to yield native-like protein conformations. While Monte Carlo sampling of only one fragment at a time allows the native structures to be smoothly constructed, the existence of probability dependencies between the different local conformations seems clear. These dependencies could be exploited during the search applying probability models in a context similar to EDAs.

We have shown that EDAs can mimic some features of the actual protein folding process. Our approach to the protein folding process in the HP simplified model has paid attention to how the order parameters such as the contact order influence on the behavior of EDAs. Such an analysis is scarce in previous nature-inspired approaches to the HP problem.

The experiments that simulate the protein folding process presented in this paper have only investigated a number of the potential issues that can be studied with the model. The analysis of EDA simulations lead to the study of other features exhibited by the model. One example is the emergence of nucleation events in protein folding.

It is generally accepted that some sort of nucleation event is a key to the protein folding mechanism. This means that some local partial structures are generally formed before the configuration of the whole protein. However, there are different ways to explain how this mechanism operates. Two opposite explanations are the "many delocalized nuclei" and the "specific nucleus" ideas. The former states that each conformation in the ensemble contains a different locally structured region. The second idea suggests that the transition state ensemble is comprised of conformations that share the same set of essential contacts, which form a compact core inside the native state (the specific nucleus) [40].

Emergence of nucleation events in protein folding can be approached by the study of the marginal probability distributions associated with the local configurations. By adding *a priori* information about the local structure configurations to the probabilistic model, the EDA protein folding model can also be harnessed to test the "specific nucleus" or "many delocalized nuclei" hypothesis.

While the application of the model to real folding problems is clearly constrained by the nature of the HP energy model and the specificities of EDAs, we recall that different applications of protein models require different levels of accuracy. Some models can be used to study catalytic mechanisms, while other are more suitable to find functional sites by 3-d motif searching [86]. Furthermore, as shown in this paper, simple models can be analyzed in detail over a wide range of instances and parameters.

Finally, we point out that measures associated with the protein folding process, such as the contact order and the degree of frustration in proteins, are of interest for the design and study of hardness measures (e.g., fitness distance correlation, epistasis variance, etc., [87]) for evolutionary algorithms.

ACKNOWLEDGMENT

The authors thank C. Cotta for providing the programming code used in the comparison between EDAs and the hybrid GA. They also thank two anonymous reviewers whose comments have contributed to improve a previous version of this paper.

REFERENCES

[1] C. Levinthal, "Are there pathways for protein folding?," *J. Chem. Phys.*, vol. 65, pp. 44–45, 1968.
[2] K. A. Dill, "Theory for the folding and stability of globular proteins," *Biochemistry*, vol. 24, no. 6, pp. 1501–1509, 1985.
[3] H. S. Chan and E. Bornberg-Bauer, "Perspectives on protein evolution from simple exact models," *Appl. Bioinformatics*, vol. 1, no. 3, pp. 121–144, 2002.
[4] G. Chikenji, Y. Fujitsuka, and S. Takada, "Shaping up the protein folding funnel by local interaction: Lesson from a structure prediction study," in *Proc. Natl. Acad. Sci. (PNAS)*, 2006, vol. 103, no. 9, pp. 3141–3146.
[5] A. Gupta, J. Manuch, and L. Stacho, "Structure-approximating inverse protein folding problem in 2D HP model," *J. Comput. Biol.*, vol. 12, no. 10, pp. 1328–1345, 2005.
[6] H. Cejtin, J. Edler, A. Gottlieb, R. Helling, H. Li, J. Philbin, N. Wingreen, and C. Tang, "Fast tree search for enumeration of a lattice model of protein folding," *J. Chem. Phys.*, vol. 116, pp. 121–144, 2002.
[7] H. Abe and H. Wako, "Analyses of simulations of three-dimensional lattice proteins in comparison with a simplified statistical mechanical model of protein folding," *Phys. Rev. E. Statist., Nonlinear, and Soft Matter Phys.*, vol. 74, p. 011913, 2006.
[8] F. P. Huard, C. M. Deane, and G. R. Wood, "Modelling sequential protein folding under kinetic control," *Bioinformatics*, vol. 22, no. 14, pp. 203–210, 2006.
[9] S. C. Kou, J. Oh, and W. H. Wong, "A study of density of states and ground states in hydrophobic-hydrophilic protein folding models by equi-energy sampling," *J. Chem. Phys.*, vol. 124, pp. 244903(1)–244903(11), 2006.
[10] P. Crescenzi, D. Goldman, C. H. Papadimitriou, A. Piccolboni, and M. Yannakakis, "On the complexity of protein folding," *J. Comput. Biol.*, vol. 5, no. 3, pp. 423–466, 1998.
[11] B. Berger and T. Leight, "Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete," *J. Comput. Biol.*, vol. 5, no. 1, pp. 27–40, 1998.
[12] W. E. Hart and S. C. Istrail, "Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal," *J. Comput. Biol.*, vol. 3, no. 1, pp. 53–96, 1996.
[13] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler, "Testing a new Monte Carlo algorithm for protein folding," *Proteins: Structure, Function, and Genetics*, vol. 32, pp. 52–66, 1998.
[14] T. C. Beutler and K. A. Dill, "A fast conformational search strategy for finding low energy structures of model proteins," *Protein Sci.*, vol. 5, pp. 2037–2043, 1996.
[15] R. König and T. Dandekar, "Improving genetic algorithms for protein folding simulations by systematic crossover," *Biosystems*, vol. 50, pp. 17–25, 1999.
[16] N. Krasnogor, W. E. Hart, J. Smith, and D. A. Pelta, "Protein structure prediction with evolutionary algorithms," in *Proc. Genetic Evol. Comput. Conf.*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., Orlando, Florida, USA, 1999, vol. 2, pp. 1596–1601.

[17] N. Krasnogor, B. Blackburne, E. K. Burke, and J. D. Hirst, "Multimeme algorithms for protein structure prediction," in *Parallel Problem Solving from Nature—PPSN VII*, ser. Lecture Notes in Computer Science, J. J. Merelo, P. Adamidis, H. G. Beyer, J. L. Fernandez-Villacañas, and H. P. Schwefel, Eds. Granada, Spain: Springer Verlag, 2002, vol. 2439, pp. 769–778.

[18] A. Shmygelska, R. A. Hernández, and H. H. Hoos, "An ant colony optimization algorithm for the 2D HP protein folding problem," in *Proc. 3rd Int. Workshop on Ant Algorithms*, 2002, pp. 40–53.

[19] R. Unger and J. Moult, "Genetic algorithms for protein folding simulations," *J. Molecular Biol.*, vol. 231, pp. 75–81, 1993.

[20] J. Blazewicz, K. Dill, P. Lukasiak, and M. Milostan, "A tabu search strategy for finding low energy structures of proteins in HP-model," *Comput. Methods in Sci. Technol.*, vol. 10, pp. 7–19, 2004.

[21] C. Cotta, "Protein structure prediction using evolutionary algorithms hybridized with backtracking," in *Artificial Neural Nets Problem Solving Methods*, ser. Lecture Notes in Computer Science, J. Mira and J. R. Alvarez, Eds. Berlin, Germany: Springer Verlag, 2003, vol. 2687, pp. 321–328.

[22] V. Cutello, G. Morelli, G. Nicosia, and M. Pavone, "Immune algorithms with aging operators for the string folding problem and the protein folding problem," in *Proc. 5th Eur. Conf. Comput. Combinatorial Optim., EvoCop-2005*, G. R. Raidl and J. Gottlieb, Eds., 2005, vol. 3448, Lecture Notes in Computer Science, pp. 80–90.

[23] V. Cutello, G. Nicosia, M. Pavone, and J. Timmis, "An immune algorithm for protein structure prediction on lattice models," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 101–117, Feb. 2007.

[24] M. D. T. Hoque, M. Chetty, and L. S. Dooley, "A new guided genetic algorithm for 2d hydrophobic-hydrophilic model to predict protein folding," in *Proc. 2005 Congr. Evol. Comput., CEC-2005*, Edinburgh, U.K., 2005, pp. 259–266.

[25] H.-P. Hsu, V. Mehra, W. Nadler, and P. Grassberger, "Growth algorithms for lattice heteropolymers at low temperatures," *J. Chem. Phys.*, vol. 118, no. 1, pp. 444–451, 2003.

[26] D. Pelta and N. Krasnogor, "Multimeme algorithms using fuzzy logic based memes for protein structure prediction," in *Recent Advances in Memetic Algorithms*, ser. Studies in Fuzziness and Soft Computing, W. H. William, N. Krasnogor, and J. E. Smith, Eds. New York: Springer, 2004, pp. 49–64.

[27] R. Santana, P. Larrañaga, and J. A. Lozano, "Protein folding in 2-dimensional lattices with estimation of distribution algorithms," in *Proc. 1st Int. Symp. Biol. Medical Data Anal.*, 2004, vol. 3337, Lecture Notes in Computer Science, pp. 388–398.

[28] A. Shmygelska and H. H. Hoos, "An improved ant colony optimization algorithm for the 2D HP protein folding problem," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, Y. Xiang and B. Chaib-draa, Eds. Berlin, Germany: Springer-Verlag, 2003, vol. 2671, pp. 400–417.

[29] J. Smith, "The co-evolution of memetic algorithms for protein structure prediction," in *Recent Advances in Memetic Algorithms*, ser. Studies in Fuzziness and Soft Computing, W. H. William, N. Krasnogor, and J. E. Smith, Eds. New York: Springer, 2004, pp. 105–128.

[30] J. Song, J. Cheng, T. T. Zheng, and J. Mao, "A novel genetic algorithm for HP model protein folding," in *Proc. 6th Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT-2005)*, 2005, pp. 935–937.

[31] G. W. Greenwood and J.-M. Shin, "On the evolutionary search for solutions to the protein folding problem," in *Evolutionary Computation in Bioinformatics*, G. B. Fogel and D. W. Corne, Eds. San Mateo, CA: Morgan Kaufmann, 2002, pp. 115–136.

[32] K. Ginalski, N. V. Grishin, A. Godzik, and L. Rychlewski, "Practical lessons from protein structure prediction," *Nucleid Acids Research*, vol. 33, no. 6, pp. 1874–1891, 2005.

[33] S. Duarte-Flores and J. E. Smith, "Study of fitness landscapes for the HP model of protein structure prediction," in *Proc. 2003 Congr. Evol. Comput., CEC-2003*, R. Sarker, R. Reynolds, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, Eds., 2003, pp. 2338–2345, IEEE Press.

[34] , P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Norwell, MA: Kluwer, 2002.

[35] , J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Berlin, Germany: Springer-Verlag, 2006.

[36] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature—PPSN IV*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1996, vol. 1141, LNCS, pp. 178–187.

[37] M. Pelikan, D. E. Goldberg, and F. Lobo, "A survey of optimization by building and using probabilistic models," *Comput. Optim. Appl.*, vol. 21, no. 1, pp. 5–20, 2002.

[38] V. S. Pande, A. Y. Grosberg, T. Tanaka, and D. S. Rokhsar, "Protein folding pathways: Is a "new view" needed?," *Current Opinion in Structural Biology*, vol. 8, no. 1, pp. 68–79, 1998.

[39] J. N. Onuchic and P. G. Wolynes, "Theory of protein folding," *Current Opinion in Structural Biology*, vol. 14, pp. 70–75, 2004.

[40] D. Baker, "A surprising simplicity to protein folding," *Nature*, vol. 405, pp. 39–42, 2000.

[41] J. E. Shea, J. N. Onuchic, and C. L. Brooks, "Exploring the origins of topological frustration: Design of a minimally frustrated model of fragment b of protein a," in *Proc. Natl. Acad. Sci. (PNAS)*, 1999, vol. 96, no. 22, pp. 12512–12517.

[42] L. Kallel, B. Naudts, and R. Reeves, "Properties of fitness functions and search landscapes," in *Theoretical Aspects of Evolutionary Computing*, L. Kallel, B. Naudts, and A. Rogers, Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 177–208.

[43] J. N. Onuchic, H. Nymeyer, A. E. Garcia, J. Chahine, and N. D. Socci, "The energy landscape theory of protein folding: Insights into folding mechanisms and scenarios," *Adv. Protein Chem.*, vol. 53, pp. 87–152, 2000.

[44] L. L. Chavez, J. N. Onuchic, and C. Clementi, "Quantifying the roughness on the free energy landscape: Entropic bottlenecks and protein folding rates," *J. Amer. Chem. Soc.*, vol. 126, no. 27, pp. 8426–8432, 2004.

[45] R. L. Dunbrack, "Rotamer libraries in the 21st century," *Current Opinion in Structural Biol.*, vol. 12, pp. 431–440, 2002.

[46] G. A. Lazar, J. R. Desjarlais, and T. M. Handel, "De novo protein design of the hydrophobic core of ubiquitin," *Protein Science*, vol. 6, pp. 1167–1178, 1997.

[47] C. M. Kraemer-Pecore, A. M. Wollacott, and J. R. Desjarlais, "Computational protein design," *Current Opinion in Chem. Biol.*, vol. 5, pp. 690–695, 2001.

[48] C. A. Rohl, C. E. M. Strauss, K. Misura, and D. Baker, "Protein structure prediction using Rosetta," *Methods in Enzymology*, vol. 383, pp. 66–93, 2004.

[49] Y. Cui, W. H. Wong, E. Bornberg-Bauer, and H. S. Chan, "Recombinatoric exploration of novel folded structures: A heteropolymer-based model of protein evolutionary landscapes," in *Proc. Natl. Acad. Sci.* , 2002, vol. 99, no. 2, pp. 809–814.

[50] V. Cutello, G. Nicosia, and M. Pavone, "An immune algorithm with hyper-macromutations for the Dill's 2d hydrophobic-hydrophilic model," in *Proc. 2004 Congr. Evol. Comput., CEC-2004*, Portland, OR, 2004, vol. 1, pp. 1074–1080.

[51] J. D. Hirst, "The evolutionary landscape of functional model proteins," *Protein Eng.*, vol. 12, pp. 721–726, 1999.

[52] N. Metropolis, A. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1091, 1953.

[53] M. Khimasia and P. Coveney, "Protein structure prediction as a hard optimization problem: The genetic algorithm approach," *Molecular Simulation*, vol. 19, pp. 205–226, 1997.

[54] U. H. E. Hansmann and Y. Okamoto, "New Monte Carlo algorithms for protein folding," *Current Opinion in Structural Biol.*, vol. 9, pp. 177–181, 1999.

[55] S. Liang and W. H. Wong, "Evolutionary Monte Carlo for protein folding simulation," *J. Chem. Phys.*, vol. 115, pp. 3374–3380, 2001.

[56] H. Frauenkron, U. Bastolla, E. Gerstner, P. Grassberger, and W. Nadler, "New Monte Carlo algorithm for protein folding," *Phys. Rev. Lett.*, vol. 80, no. 4, pp. 3149–3153, 1998.

[57] H.-P. Hsu, V. Mehra, and P. Grassberger, "Structure optimization in an off-lattice protein model," *Phy. Rev. E*, vol. 68, no. 2, p. 4, 2003.

[58] P. Grassberger, "Sequential Monte Carlo methods for protein folding," in *Proc. NIC Symp. 2004*, D. Wolf, G. Münster, and M. Kremer, Eds., 2004, vol. 20, John von Neumann-Institut für Computing (NIC), pp. 1–10.

[59] C. A. Voigt, D. B. Gordon, and S. L. Mayo, "Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design," *J. Molecular Biol.*, vol. 299, no. 3, pp. 799–803, 2000.

[60] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles, "Machine learning in bioinformatics," *Briefings in Bioinformatics*, vol. 7, pp. 86–112, 2006.

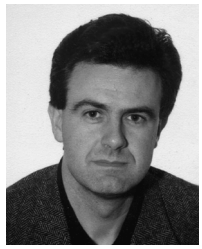
[61] R. Santana, P. Larrañaga, and J. A. Lozano, "Side chain placement using estimation of distribution algorithms," *Artif. Intell. Med.* vol. 39, no. 1, pp. 49–63, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6T4K-4KF785F-1/2/d65e9d9d9552ef31ec57d7dca349a4fc

[62] R. Santana, P. Larrañaga, and J. A. Lozano, "Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem," *J. Heuristics*, 2007, accepted for publication.

[63] A. Schug and W. Wenzel, "An evolutionary strategy for all-atom folding on the 60-amino-acid bacterial ribosomal proten L20," *Biophys. J.*, vol. 9, pp. 4273–4280, 2006.
[64] J. T. Pedersen and J. Moult, "Protein folding simulation with genetic algorithms and a detailed molecular description," *J. Molecular Biol.*, vol. 269, pp. 240–259, 1997.
[65] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space," in *Proc. 14th Int. Conf. Mach. Learning*, 1997, pp. 30–38.
[66] H. Mühlenbein, T. Mahnig, and A. Ochoa, "Schemata, distributions and graphical models in evolutionary optimization," *J. Heuristics*, vol. 5, no. 2, pp. 213–247, 1999.
[67] P. Larrañaga, "An introduction to probabilistic graphical models," in *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Norwell, MA: Kluwer, 2002, pp. 25–54.
[68] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.
[69] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287–297, Nov. 1999.
[70] J. S. De Bonet, C. L. Isbell, and P. Viola, "MIMIC: Finding optima by estimating probability densities," in *Adv. Neural Inf. Process. Syst.*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, vol. 9, pp. 424–430.
[71] G. Harik, "Linkage learning via probabilistic modeling in the EcGA," Univ. Illinois at Urbana–Champaign, Illinois Genetic Algorithms Lab., Urbana, IL, IlliGAL Rep. 99010, 1999.
[72] R. Etxeberria and P. Larrañaga, "Global optimization using Bayesian networks," in *Proc. 2nd Symp. Artif. Intell. (CIMAF-99)*, A. Ochoa, M. R. Soto, and R. Santana, Eds., Habana, Cuba, 1999, pp. 151–173.
[73] M. Pelikan, D. E. Goldberg, and E. Cantü-Paz, "BOA: The Bayesian optimization algorithm," in *Proc. Genetic and Evol. Comput. Conf., GECCO-1999*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., Orlando, FL, 1999, pp. 525–532.
[74] H. Mühlenbein and T. Mahnig, "Evolutionary synthesis of Bayesian networks for optimization," in *Advances in Evolutionary Synthesis of Intelligent Agents*, M. Patel, V. Honavar, and K. Balakrishnan, Eds. Cambridge, MA: MIT Press, 2001, pp. 429–455.
[75] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 462–467, May 1968.
[76] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Proc. 2nd Ann. Conf. Uncertainty in Artif. Intell.*, J. F. Lemmer and L. N. Kanal, Eds., 1988, pp. 149–164.
[77] M. Meila, "Learning mixtures of trees," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1999.
[78] R. Santana, A. Ochoa, and M. R. Soto, "The mixture of trees factorized distribution algorithm," in *Proc. Genetic and Evol. Comput. Conf., GECCO-2001*, L. Spector, E. Goodman, A. Wu, W. Langdon, H. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, Eds., San Francisco, CA, 2001, pp. 543–550.
[79] N. Lesh, M. Mitzenmacher, and S. Whitesides, "A complete and effective move set for simplified protein folding," Mitsubishi Electric Res. Lab., Tech. Rep. TR-2003-03, Feb, 2003.
[80] R. Bonneau, I. Ruczinski, J. Tsai, and D. Baker, "Contact order and ab initio protein structure prediction," *Protein Science*, vol. 11, pp. 1937–1944, 2002.
[81] R. Santana, P. Larrañaga, and J. A. Lozano, "The role of *a priori* information in the minimization of contact potentials by means of estimation of distribution algorithms," in *Proc. 5th Eur. Conf. Evol. Comput., Mach. Learn. Data Mining in Bioinformatics*, E. Marchiori, J. H. Moore, and J. C. Rajapakse, Eds., 2007, vol. 4447, Lecture Notes in Computer Science, pp. 247–257.
[82] R. Blanco, P. Larrañaga, I. Inza, and B. Sierra, "Selection of highly accurate genes for cancer classification by estimation of distribution algorithms," in *Proc. Workshop Bayesian Models in Medicine Held within AIME 2001*, 2001, pp. 29–34.
[83] Y. Saeys, S. Degroeve, D. Aeyels, Y. Van de Peer, and P. Rouzé, "Fast feature selection using a simple estimation of distribution algorithm: A case study on splice site prediction," *Bioinformatics*, vol. 19, no. 2, pp. ii179–ii188, 2003.
[84] Y. Saeys, S. Degroeve, D. Aeyels, P. Rouzé, and Y. Van de Peer, "Feature selection for splice site prediction: A new method using EDA-based feature ranking," *BMC Bioinformatics*, vol. 5, pp. 64–75, 2004.
[85] J. Peña, J. A. Lozano, and P. Larrañaga, "Unsupervised learning of Bayesian networks via estimation of distribution algorithms: An application to gene expression data clustering," *Int. J. Uncertainty, Fuzziness and Knowledge-Based Syst.*, vol. 12, no. 1, pp. 63–82, 2004.
[86] D. Baker, "Protein structure prediction and structural genomics," *Science*, vol. 294, pp. 93–96, 2001.
[87] B. Naudts and L. Kallel, "A comparison of predictive measures of problem difficulty in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 1, pp. 1–15, Apr. 2000.

Roberto Santana received the B.S. degree in computer science and the Ph.D. degree in mathematics from the University of Havana, Havana, Cuba, in 1996 and 2005, respectively, and the Ph.D. degree in computer science from the University of the Basque Country, Guipuzcoa, Spain, in 2006.

He is a Postdoctoral Researcher at the University of the Basque Country. His main research interests are evolutionary computation, probabilistic graphical models, and bioinformatics.



Pedro Larrañaga received the M.S. degree in mathematics from the University of Valladolid, Valladolid, Spain, in 1981, and the Ph.D. degree in computer science from the University of the Basque Country, Guipuzcoa, Spain, in 1995.

He is a Professor of Computer Science and Artificial Intelligence at the Technical University of Madrid. He has published over 40 refereed journal papers. His main research interests are in the areas of evolutionary computation, machine learning, probabilistic graphical models, and bioinformatics.

José A. Lozano (M'04) received the B.S. degrees in mathematics and computer science and the Ph.D. degree from the University of the Basque Country, Guipuzcoa, Spain, in 1991, 1992, and 1998, respectively.

Since 1999, he has been an Associate Professor of Computer Science at the University of the Basque Country. He has edited three books and has published over 25 refereed journal papers. His main research interests are evolutionary computation, machine learning, probabilistic graphical models, and bioinformatics.