Artificial Intelligence 212 (2014) 59–79

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Computational protein design as an optimization problem ☆

David Allouche [a], Isabelle André [b,c,d], Sophie Barbe [b,c,d], Jessica Davies [a], Simon de Givry [a], George Katsirelos [a], Barry O'Sullivan [e], Steve Prestwich [e], Thomas Schiex [a,*], Seydou Traoré [b,c,d]

[a] MIAT, UR-875, INRA, F-31320 Castanet Tolosan, France
[b] Université de Toulouse; INSA, UPS, INP; LISBP, 135 Avenue de Rangueil, F-31077 Toulouse, France
[c] CNRS, UMR5504, F-31400 Toulouse, France
[d] INRA, UMR792, Ingénierie des Systèmes Biologiques et des Procédés, F-31400 Toulouse, France
[e] Insight Centre for Data Analytics, University College Cork, Ireland

ARTICLE INFO

Article history:
Received 16 September 2013
Received in revised form 22 February 2014
Accepted 10 March 2014
Available online 26 March 2014

Keywords:
Weighted constraint satisfaction problem
Soft constraints
Neighborhood substitutability
Constraint optimization
Graphical model
Cost function networks
Integer linear programming
Quadratic programming
Computational protein design
Bioinformatics
Maximum a posteriori inference
Maximum satisfiability

ABSTRACT

Proteins are chains of simple molecules called amino acids. The three-dimensional shape of a protein and its amino acid composition define its biological function. Over millions of years, living organisms have evolved a large catalog of proteins. By exploring the space of possible amino acid sequences, protein engineering aims at similarly designing tailored proteins with specific desirable properties. In Computational Protein Design (CPD), the challenge of identifying a protein that performs a given task is defined as the combinatorial optimization of a complex energy function over amino acid sequences.

In this paper, we introduce the CPD problem and some of the main approaches that have been used by structural biologists to solve it, with an emphasis on the exact method embodied in the dead-end elimination/A^* algorithm (DEE/A^*). The CPD problem is a specific form of binary Cost Function Network (CFN, aka Weighted CSP). We show how DEE algorithms can be incorporated and suitably modified to be maintained during search, at reasonable computational cost.

We then evaluate the efficiency of CFN algorithms as implemented in our solver `toulbar2`, on a set of real CPD instances built in collaboration with structural biologists. The CPD problem can be easily reduced to 0/1 Linear Programming, 0/1 Quadratic Programming, 0/1 Quadratic Optimization, Weighted Partial MaxSAT and Graphical Model optimization problems. We compare `toulbar2` with these different approaches using a variety of solvers. We observe tremendous differences in the difficulty that each approach has on these instances.

Overall, the CFN approach shows the best efficiency on these problems, improving by several orders of magnitude against the exact DEE/A^* approach. The introduction of dead-end elimination before or during search allows to further improve these results.

☆ This paper is an invited revision of a paper which first appeared at the CP-2012 conference.

* Corresponding author.
E-mail address: Thomas.Schiex@toulouse.inra.fr (T. Schiex).

http://dx.doi.org/10.1016/j.artint.2014.03.005
0004-3702/

1. Introduction

A protein is a sequence of basic building blocks called *amino acids*. Proteins are involved in nearly all structural, catalytic, sensory, and regulatory functions of living systems [26]. Performing these functions generally requires that proteins are assembled into well-defined three-dimensional structures specified by their amino acid sequence. Over millions of years, natural evolutionary processes have shaped and created proteins with novel structures and functions by means of sequence variations, including mutations, recombinations and duplications. Protein engineering techniques coupled with high-throughput automated procedures make it possible to mimic the evolutionary process on a greatly accelerated timescale, and thus increase the odds to identify the proteins of interest for technological uses [71]. This holds great interest for medicine, synthetic biology, nanotechnologies and biotechnologies [67,75,39]. In particular, protein engineering has become a key technology to generate tailored enzymes able to perform novel specific transformations under specific conditions. Such biochemical transformations enable to access a large repertoire of small molecules for various applications such as biofuels, chemical feedstocks and therapeutics [45,11]. The development of enzymes with required substrate selectivity, specificity and stability can also be profitable to overcome some of the difficulties encountered in synthetic chemistry. In this field, the *in vitro* use of artificial enzymes in combination with organic chemistry has led to innovative and efficient routes for the production of high value molecules while meeting the increasing demand for ecofriendly processes [61,13]. Nowadays, protein engineering is also being explored to create non-natural enzymes that can be combined in vivo with existing biosynthetic pathways, or be used to create entirely new synthetic metabolic pathways not found in nature to access novel biochemical products [28]. These latest approaches are central to the development of synthetic biology. One significant example in this field is the full-scale production of the antimalarial drug (artemisinin) from the engineered bacteria *Escherichia coli* [66].

With a choice among 20 naturally occurring amino acids at every position, the size of the combinatorial sequence space is out of reach for current experimental methods, even for short sequences. Computational protein design (CPD) methods therefore try to intelligently guide the protein design process by producing a *collection* of proteins, that is rich in functional proteins, but small enough to be experimentally evaluated. The challenge of choosing a sequence of amino acids to perform a given task is formulated as an optimization problem, solvable computationally. It is often described as the inverse problem of protein folding [70]: the three-dimensional structure is known and we have to find amino acid sequences that fold into it. It can also be considered as a highly combinatorial variant of side-chain positioning [82] because of possible amino acid mutations.

Various computational methods have been proposed over the years to solve this problem and several success stories have demonstrated the outstanding potential of CPD methods to engineer proteins with improved or novel properties. CPD has been successfully applied to increase protein thermostability and solubility; to alter specificity towards some other molecules; and to design various binding sites and construct *de novo* enzymes (see for example [46]).

Despite these significant advances, CPD methods must still mature in order to better guide and accelerate the construction of tailored proteins. In particular, more efficient computational optimization techniques are needed to explore the vast combinatorial space, and to facilitate the incorporation of more realistic, flexible protein models. These methods need to be capable of not only identifying the optimal model, but also of enumerating solutions close to the optimum.

We begin by defining the CPD problem with rigid backbone, and then introduce the approach commonly used in structural biology to exactly solve CPD. This approach relies on dead-end elimination (DEE), a specific form of dominance analysis that was introduced in [24], and later strengthened in [37]. If this polynomial-time analysis does not solve the problem, an A^* algorithm is used to identify an optimal protein design.

We observe that the rigid backbone CPD problem can be naturally expressed as a Cost Function Network (aka Weighted Constraint Satisfaction Problem). In this context, DEE is similar to neighborhood substitutability [27]. We show how DEE can be suitably modified so as to be maintained during search at reasonable computational cost, in collaboration with the usual soft local consistencies.

To evaluate the efficiency of the CFN approach, we model the CPD problem using several combinatorial optimization formalisms. We compare the performance of the 0/1 linear programming and 0/1 quadratic programming solver `cplex`, the semidefinite programming based Boolean quadratic optimization tool `biqmac`, several weighted partial MaxSAT solvers, the Markov random field optimization solvers `daoopt` and `mplp` [80], and the CFN solver `toulbar2`, against that of a well-established CPD approach implementing DEE/A^*, on various realistic protein design problems. We observe drastic differences in the difficulty that these instances represent for different solvers, despite often closely related models and solving techniques.

2. The computational protein design approach

A protein is a sequence of organic compounds called amino acids. All amino acids consist of a common *peptidic core* and a *side chain* with varying chemical properties (see Fig. 1). In a protein, amino acid cores are linked together in sequence to form the *backbone* of the protein. A given protein *folds* into a 3D shape that is determined from the sequence of amino acids. Depending upon the amino acid considered, the side chain of each individual amino acid can be rotated along up to 4 dihedral angles relative to the backbone. After Anfinsen's work [3], the 3D structure of a protein can be considered to be defined by the backbone and the set of side-chain rotations. This is called the *conformation* of the protein and it determines its chemical reactivity and biological function.