Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Naïve Bayes ant colony optimization for designing high dimensional experiments

Q1 M. Borrotti[a,b,*], G. Minervini[d], D. De Lucrezia[e], I. Poli[b,c]

[a] Institute of Applied Mathematics and Information Technologies, CNR-IMATI, via Bassini 15, 20133 Milan, Italy
[b] European Centre for Living Technology, Ca' Foscari University of Venice, San Marco 2940, 30124 Venice, Italy
[c] Department of Environmental Science, Informatics and Statistics, Ca' Foscari University of Venice, Dorsoduro 2137, 30123 Venice, Italy
[d] Department of Biology, University of Padua, Via U. Bassi 58, 35121 Padua, Italy
[e] Explora Biotech S.r.l., Via della Libertá 9, 30175 Venice, Italy

ARTICLE INFO

*Article history:*
Received 15 October 2015
Received in revised form 8 July 2016
Accepted 10 August 2016
Available online xxx

*Keywords:*
Ant colony optimization
Naïve Bayes classifier
Experimental design
Enzyme engineering
High dimensionality

ABSTRACT

In a large number of experimental problems, high dimensionality of the search area and economical constraints can severely limit the number of experimental points that can be tested. Within these constraints, classical optimization techniques perform poorly, in particular, when little a priori knowledge is available. In this work we investigate the possibility of combining approaches from statistical modeling and bio-inspired algorithms to effectively explore a huge search space, sampling only a limited number of experimental points. To this purpose, we introduce a novel approach, combining ant colony optimization (ACO) and naïve Bayes classifier (NBC) that is, the naïve Bayes ant colony optimization (NACO) procedure. We compare NACO with other similar approaches developing a simulation study. We then derive the NACO procedure with the goal to design artificial enzymes with no sequence homology to the extant one. Our final aim is to mimic the natural fold of 200 amino acids 1AGY serine esterase from *Fusarium solani*.

## 1. Introduction

Q3 Enzymes are biological molecules that catalyze thousands of chemical reactions that sustain life. Each enzyme is a polymer generated from a series of up to 20 different amino acids. Polymers can be represented as words formed according to the alphabet $\mathbf{a} = \{a_1, a_2, a_3, \ldots, a_{20}\}$. They may differ in length, sequence and amino acid order. A secondary and tertiary structure, which defines enzyme activity, is associated to each sequence [1]. Enzyme activities can be associated to each of these combinations of amino acids through experimentation. Biological experimentation aims to optimize the sequence of amino acids in order to find enzymes with the best activities. The number of possible combinations of amino acids rapidly increase leading to a combinatorial explosion of the search space. Designing artificial enzymes with desired properties is known in computational biology as *Enzyme Engineering* [2,3]. Enzyme Engineering is an important task in numerous fields such as chemicals, pharmaceuticals, fuel, food or agricultural additives [4].

In this context it is necessary to develop new enzyme engineering procedures based on computational approaches able to identify best solutions with the lowest number of costly and time consuming experimentations. Different optimization approaches have been proposed in the literature to tackle problems with high dimensional and complex search spaces [5–11]. Some approaches are based on Evolutionary Computation Algorithms [12]. Genetic Algorithms (GAs) [13,14], for example, use techniques inspired by natural evolution, i.e. mutation, selection and crossover, in order to optimize complex functions. The application of these approaches is limited by several issues such as the need of solid *a priori* knowledge of the relationship between the structure and function of an enzyme, the vast number of possible modifications of an existing enzyme, the lack of efficient screening procedures able to reduce the number of modified enzymes to be tested and the computational effort needed to create new enzymes. A promising solution is combining metaheuristic algorithms and statistical models. This is a novel research area that addresses problems characterized by large design space, high-order interactions between variables, and, complex and non-linear experimental surfaces [15]. Some examples of these hybridized methods can be found in [16–21]. A recent

[*] Corresponding author at: Institute of Applied Mathematics and Information Technologies, CNR-IMATI, via Bassini 15, 20133 Milan, Italy. Q2
*E-mail address:* matteo.borrotti@mi.imati.cnr.it (M. Borrotti).

http://dx.doi.org/10.1016/j.asoc.2016.08.018
1568-4946/© 2016 Elsevier B.V. All rights reserved.

approach is proposed in [21] where an optimization algorithm, called Co-Information Composite Likelihood (COIL), based on the evolutionary paradigm is introduced by coupling cross-entropy sampling [22] and composite likelihood principles [23] to design novel enzymes with improved functionalities.

In this paper, we introduce a novel approach that combines ant colony optimization (ACO) [24–26] and the naïve Bayes Classifier (NBC) [27], called *naïve-Bayes ant colony optimization* (NACO). The ACO approach has been widely used in many optimization problems ranging from sequential ordering studies [28] to open shop scheduling [29]. ACO has been also applied to biological problems such as DNA sequencing problems [30,31].

Several theoretical and practical features of NBC have been studied that have shown its reliability in terms of classification accuracy [32–37]. One of the most powerful procedures proposed in the biological field is an algorithm for genetic biomarker selections and subject classifications from the simultaneous analysis of genome-wide Single-Nucleotide Polymorphism (SNP) data based on the naïve Bayes framework [37].

The key idea of NACO is to include information achieved via the NBC method in the path construction mechanism of ACO, in order to drive the search process towards the target region in a much more effective way. NACO was evaluated both in a simulation study and in a real case of building artificial enzymes. In particular, the NACO algorithm was constructed and a simulation comparison with a set of other procedures frequently used for this purpose was developed. The algorithm was then derived with the goal of designing artificial enzymes with no sequence homology to extant ones; the final aim being to mimic the natural fold of 200 amino acid long 1AGY serine esterase from *Fusarium solani* [38].

This paper is organized as follows. In Section 2 the optimization problem is formalized and in Section 3, we briefly describe the ACO and NBC algorithm. In Section 4, we introduce our approach, naïve-Bayes ant colony optimization (NACO). In Section 5, we evaluate NACO performance in a simulation study, comparing results with well-known techniques and, in Section 6, we apply NACO to the design of new enzymes. Finally, some conclusions are drawn in Section 7.

## 2. The optimization problem

Enzyme engineering can be described as a combinatorial optimization problem $P = (\mathcal{X}, f)$ in which:

- $f$ is the objective function to be optimized and defined as $f : D_1 \times D_2 \times \cdots \times D_d \rightarrow \mathbb{R}^+$ where $D_1, \ldots, D_d$ are the domains of the following variables in $X = \{x_1, \ldots, x_d\}$;
- $\mathcal{X}$ is the set of all possible combinations $\mathbf{x} = (x_{1l_i}, \ldots, x_{dl_i})$ where $x_{il_i} \in D_i$ indicates the value of the variable $x_i \, \forall i$ where $i = 1, \ldots, d$.

The function $f$ calculates a response value for each combination of variables, $y = f(x_{1l_i}, \ldots, x_{dl_i})$ with $l_i = 1, \ldots, k$. Within this setting, the final aim is to find the combination, or solution, $\mathbf{x}^* \in \mathcal{X}$ with maximum response value, that is, $f(\mathbf{x}^*) > f(\mathbf{x}) \forall \mathbf{x} \in \mathcal{X}$.

To address the enzyme engineering problem we propose that the variables represent the set of amino-acids in the enzyme sequence, the variable domain is the alphabet $\mathbf{a}$, $\mathcal{X}$ is the set of all possible candidate enzymes to be tested and the objective function is the enzyme activity to be maximized.

## 3. Ant colony optimization and naïve Bayes classifier

In this section we briefly introduce ant colony optimization (ACO) and naïve Bayes classifier (NBC), which we intend to combine in an new approach to high dimensional optimization problems.

### 3.1. Ant colony optimization (ACO)

Ant colony optimization (ACO) [24] is a metaheuristic algorithm introduced for complex combinatorial problems. It is inspired by the behaviour of specific real ant colonies in search of food in their environment. In nature ants explore the surrounding environment for food source through random walks. Once found the food, ants return to their colony leaving behind a trail of pheromones. Pheromones are volatile compounds that quickly dry off so that a short path from colony to food source is more likely to preserve detectable amount of pheromones. This in turn will allow more ants to follow that path and thus reinforce the pheromone trails. This positive feedback eventually leads all ants to use a single path from the colony to the food.

The main idea of the ACO algorithm is to mimic this behaviour with *artificial ants* walking around a graph representing the environment; more specifically, the problem to solve is illustrated in Fig. 1.

ACO algorithm is applied to problems that can be described as a graph $G = (N, A)$ where $N$ is the set of nodes and $A$ the set of arcs that fully connect the nodes. A weight $\tau_{i,j}$, called pheromone value, is associated with each arc which connects node $i$ with node $j$. The pheromone value represents the attractiveness of a specific arc for the ants: the higher the amount of pheromone on an arc, the higher the probability that ants will choose it when constructing solutions. In addition, a heuristic value $\eta_{i,j}$, which represents *a priori* information, is introduced to move from node $i$ to node $j$.

The construction of the ACO algorithm includes the following phases [26,39]: (i) *Construct ant solutions*, (ii) *Daemon actions* and (iii) *Update pheromone*. The first phase, *Construct ant solutions*, represents the procedure needed for ants to construct solutions incrementally: starting from a first node, ants add a new node at each iteration of the procedure in order to build the entire solution. An ant decides where to go next in accordance with pheromone values $\tau_{i,j}$ and the heuristic values $\eta_{i,j}$. The relative influence of these two values is weighed in accordance to two parameters, called $\alpha > 0$ and $\beta \geq 0$, respectively. The second phase, *Daemon actions*, comprises all problem-specific operations that may be considered for boosting the performance of ACO algorithms. The main example of such operations is the introduction of local search techniques [40,41]. The third phase, *Update pheromone*, includes the procedure to updates pheromone values. This procedure is divided in two phases. First, pheromone evaporation is applied to decrease pheromone values. The degree of decrease depends on the parameter $\rho \in [0, 1]$, called the evaporation rate. The aim of pheromone

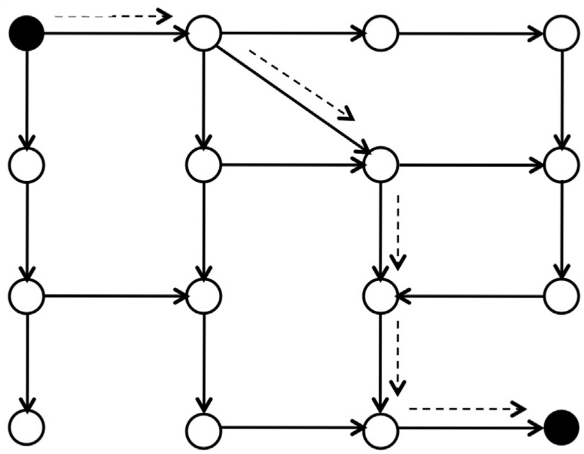

**Fig. 1.** Representation of a possible graph where ants move, from a starting node to a destination node.

Please cite this article in press as: M. Borrotti, et al., Naïve Bayes ant colony optimization for designing high dimensional experiments, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.018