An Evolutionary Approach to Experimental Design for Combinatorial Optimization with an Application to Enzyme Engineering

Matteo Borrotti ¹ and Alessandra Giovagnoli ²

¹ European Centre for Living Technology
University Ca’ Foscari of Venice
Venice, Italy

² Department of Statistical Sciences
University of Bologna
Bologna, Italy

September 2, 2011
Outline

1) The Biological Problem

2) “Large p, Small n”

3) Ant Colony Optimization

4) Evolutionary Model-Based Experimental Design
   - Model-Based Ant Colony Design (MACD)

5) Testing the Optimization Method

6) The Real-Life Application

7) Conclusions
The Biological Approach

*Enzyme Engineering and Design:* to design novel or improved enzymes

(1) An *amino-acid* is a molecule commonly identified by a letter. There are 20 amino-acids (aa) in nature.

(2) A *polypeptide* is a sequence of amino-acids.

(3) A *protein* is a polypeptide capable of folding into a three-dimensional structure.

(4) Proteins can be divided into *domains.* The minimum length required for a domain to fold is believed to be 50 amino-acids.

(5) In our case we use *artificial domains* called *pseudo-domains* (50 aa).

(6) An *enzyme* is a protein capable of catalysis, *i.e.* able to enhance chemical reactions within a cell.

*Score = A fitness function of the enzyme*
In our case, the score is measured in the range 0 - 1000.
The Case Study

Aim:
Starting with a library of 95 randomly chosen pseudo-domains
Import Feature: the amino-acid frequency reflects the composition of natural enzymes.

Generate synthetic enzymes of 4 pseudo-domains (200 amino-acids)
Example:

\[
\begin{align*}
TETTSFLITKFSPDQQLIIFQGDGYTTKEKLTLTKAVKNTVGRALYSSPI \\
HIWDETRGNVANFVTSTFTVINAPNSYNTADGFITFFIAVPVDTKPTTGGY \\
LGVFNSAEYDKTTQTVAVEFDTFYNAAWDPSNRDHRIGIDVNSIKSVNTK \\
TETTSFLITKFSPDQQLIIFQGDGYTTKEKLTLTKAVKNTVGRALYSSPI
\end{align*}
\]

Permutations of 95 elements in 4 positions (with repetitions)

\[95^4 = 8.154 \times 10^7\]

The data is the score of the evaluated enzymes

Main Challenge

Develop methodologies to construct non-natural enzymes with a specific function without the need to screen a large number of sequences in a combinatorial enzyme space.
Our problem is characterized by

\( p \) : number of variables involved in the experiment.

\( p \) at least 380 (95 domains x 4 positions)

\( n \) : number of experimental units on which data are available.

\( n = 96 \) (dimension of the well-plate)

\( n << p \) “large \( p \), small \( n \)”

- Large number of variables;
- Enormous number of different ways in which pseudo-domains can be composed in 4 positions;
- Possible network of interactions between them
- Rigid laboratory protocols.

We need to study more accurate approaches for discrete problems:
1) *statistical model for high dimensionality*: statistical models able to get as much information as possible from few data points
2) *combinatorial optimization*: optimization techniques able to move in a complex search
Possible Approaches in the Literature

Statistical Models for High Dimensionality

• Ridge Regression [Hoerl, 1962]
• Least Absolute Shrinkage and Selection Operator (LASSO) [Tibshirani, 1996]
• Elastic Net Model [Zou and Hastie, 2005]
• Sparse Additive Model (SpAM) [Ravikumar et al., 2009]

Another viewpoint: Bayesian Theory

Bayesian Network [Pearl, 1988]
Naïve Bayes Approach [Mitchell, 2005]

Metaheuristic Algorithm for Combinatorial Optimization

Population-Based Search

• Evolutionary Computation [Aslock, 2006]
• Genetic Algorithm [Goldberg, 1989]
• Ant Colony Optimization [Dorigo et al., 2004]

Single-Point Search

• Tabu Search [Glover et al., 1997]
• Simulated Annealing [Kirkpatrick et al., 1983]
**Evolutionary Model-Based Experimental Design**

*Idea:* combination of different approaches from Statistical Modeling and Optimization Algorithms.

**Ultimate aim of this work is to**
- Test the possibility of exploiting bio-inspired algorithms combined with advanced statistical techniques to search for an optimum in a discrete sequence space;
- Develop methods to solve problems where the number of variables increases and to tackle the high complexity of the system.

**Our proposed solution**

*Model-Based Ant Colony Design (MACD)*

based on:
- Closed-Loop Evolution
- Ant Colony Optimization (ACO)
Ant Colony Optimization (ACO) is a population-based, general-purpose stochastic search technique for the solution of difficult combinatorial problems, which is inspired by the pheromone trail laying and following foraging behaviour of some real ant species.
How Does Ant Colony Optimization Work?

Ant, in element \( i \), chooses the next element (\( j \) or \( y \)) in a probabilistic way according to the pheromone’s weight and the heuristic information on the connections between nodes (arches).

More formally

\[
a_{ij} = \frac{[\tau_{ij}(t)][\eta_{ij}]}{\sum_{i \in N_i} [\tau_{ij}(t)][\eta_{ij}]} \quad \forall j \in N_i
\]

where

\( \tau_{ij}(t) \): amount of “pheromone” in arc \((i, j)\) at time \( t \)

\( \eta_{ij} \): heuristic value of moving from node \( i \) to node \( j \). Generally, \textit{prior} information on the problem.

Probability of ant \( k \) moving from node \( i \) to node \( j \)

\[
p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{i \in N_i^k} a_{ij}(t)}
\]

Solution: path from the source to destination node.
procedure ACO_MetaHeuristic

    while(not_termination)
        generateSolutions()
        daemonActions()
        pheromoneUpdate()
    end while

end procedure
**MAX-MIN Ant System**

Proposed by Stützle and Hoos in the 2000.

The MAX-MIN Ant System has three important features:

(i) The ant that has reached the best solution (*i.e.* path) in the current iteration (*iteration-best ant*) deposits “pheromone”, namely some weight, after each iteration. Alternatively, the ant that found the best solution from the beginning of the trial (*global-best ant*);

(ii) The range of possible “pheromone trails” on each solution component is limited to an interval $[\tau_{\text{min}}, \tau_{\text{max}}]$;

(iii) The “pheromone trail” is deliberately initialized to $\tau_{\text{max}}$. 
Set-up
A mixed approach, exploiting both information from

- Statistical modelling (linear model with binary predictive variables, estimated by LS from data)
- Ant Colony Algorithm (MAX-MIN Ant System)

Important
*Switches between real and simulated observations*

We require that MACD will

1. Simulate the problem and move in the search space as many times as we want.
2. Get an iterated refinement of the predictive model.
Model-Based Ant Colony Design (MACD)

Learning Phase

Library of Proteins

Is Max Number of Trials reached?

No

Predictive Model

Ant Colony Optimization

Initial Random Trial of Real Proteins

Simulative Model Based Phase

Prediction Candidate Proteins

Apply Local Search around the best (i.e. SA) – if needed

Update Probability Matrix

Stop Criterion Satisfied?

No

Yes

Real Evaluation

New Trial of Proteins

End

13
Simulative test functions $\rightarrow f$

1) Polynomial Regression Model (PRM)

2) Polynomial Sparse Regression Model (PSRM)

At each response: $\epsilon \sim N(\mu = 0, \sigma^2 = 1)$

We test:

- \textit{MAX-MIN} Ant System

- \textit{Model-Based Ant Colony Design (MACD)}

Setting:
• Experimental Batches: 30
• Experiments for batch: 100
Results with the Polynomial Regression Model

Average Maximum Value over 100 simulations

Solid Line: MACD
Dashed Line: MMAS

95% Confidence intervals are plotted around the average values
Results with the Polynomial Sparse Regression Model

Average Maximum Value over 100 simulations

Solid Line: MACD
Dashed Line: MMAS

95% Confidence intervals are plotted around the average values
The Real-Life Application

Real Experiment

- Library of 95 randomly chosen pseudo-domains
- Generate synthetic enzymes of 4 pseudo-domains
- Search Space of $95^4$ candidate enzymes
- Initial data set of 96 randomly chosen enzymes
**Preliminary Remarks**

- Number of trials: 6
  In order to be competitive, desirable number of trials is 5
- Number of enzymes tested at each trial: 96

**Rule:**

At step $i$ the best solution is compared with the best solution at step $i-1$. If the best solution at step $i-1$ is not incremented by at least 4% then all the probabilities $p_{ij}$ are set to a uniform distribution. This rule is applied starting from $i = 3$. 
Distribution of the Score of trial # 1

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial #1</td>
<td>242.0</td>
<td>322.2</td>
<td>375.0</td>
<td>381.4</td>
<td>433.5</td>
<td>696.0</td>
<td>83.9</td>
</tr>
</tbody>
</table>

Score Distribution Trial 1
$p_{ij}$ are set to a uniform distribution
The last trial

Time for a new trial: 40 minutes.

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial #1</td>
<td>242.0</td>
<td>322.2</td>
<td>375.0</td>
<td>381.4</td>
<td>433.5</td>
<td>696.0</td>
<td>83.9</td>
</tr>
<tr>
<td>Trial #2</td>
<td>410.0</td>
<td>505.5</td>
<td>554.5</td>
<td>546.7</td>
<td>594.0</td>
<td>724.0</td>
<td>68.3</td>
</tr>
<tr>
<td>Trial #3</td>
<td>444.0</td>
<td>575.5</td>
<td>575.5</td>
<td>575.9</td>
<td>603.8</td>
<td>745.0</td>
<td>49.2</td>
</tr>
<tr>
<td>Trial #4</td>
<td>256.0</td>
<td>447.5</td>
<td>447.5</td>
<td>446.1</td>
<td>511.2</td>
<td>647.0</td>
<td>85.4</td>
</tr>
<tr>
<td>Trial #5</td>
<td>539.0</td>
<td>589.5</td>
<td>589.5</td>
<td>599.1</td>
<td>625.0</td>
<td>756.0</td>
<td>44.4</td>
</tr>
<tr>
<td>Trial #6</td>
<td>537.0</td>
<td>579.5</td>
<td>604.0</td>
<td>616.1</td>
<td>635.5</td>
<td>834.0</td>
<td>52.8</td>
</tr>
</tbody>
</table>

$p_j$ are set to a uniform distribution
MACD allows us to:

- Create a dialogue between design and laboratory experimentation at each trial, detecting a path in the combinatorial search space that leads toward a region of optimality.

- Test a small number of candidate solutions, thus making the experimental phase faster and more effective.

Furthermore, the approach has shown:

- Their ability to identify new enzymes in a very large search space of competitive candidates.

- A remarkable shift of the initial population towards higher response value areas of the search space.
**Future Work**

- Using different predictive models in MACD;
- Implementing an automatic selection of statistical models;
- Treat the relevant information contained in each trial in a different way with respect to classical optimization algorithms;
- Testing MACD in different artificial benchmark problems;
- Studying the observed behaviour of the algorithms more in depth.

**In the real experimentation:**

- Evaluation of the trials needed to reach a satisfactory solution;
- Generalizing MACD to extend the fields of applicability.
**Future Work**

- Using different predictive models in MACD;
- Implementing an automatic selection of statistical models;
- Treat the relevant information contained in each trial in a different way with respect to classical optimization algorithms;
- Testing MACD in different artificial benchmark problems;
- Studying the observed behaviour of the algorithms more in depth.

**In the real experimentation:**

- Evaluation of the trials needed to reach a satisfactory solution;
- Generalizing MACD to extend the fields of applicability.
Naïve Bayes Ant Colony Optimization (NACO)

Set-up
• Naïve Bayes Classifier
• Ant Colony Algorithm (*MAX*-MIN Ant System)

Important
Better pseudo-domains are identified in each positions and the interactions between pseudo-domains in a sequence are captured.

We require that NACO will

1. Improve upon the limits of the individual techniques and be able to deal with a large experimental space of possible solutions.
2. Not be computationally intensive.
3. Allow the researcher to be fast in creation and analysis of candidate enzymes.
How NACO Works

(a) Previous State of the Graph

(b) Naive Bayes Classification

(c) Iteration Best Solution

Next State of the Graph

\[ a_{ij} = \frac{[\tau_{ij}(t)][\eta_{ij}][\lambda_{ij}]}{\sum_{l \in N_j} [\tau_{ij}(t)][\eta_{ij}][\lambda_{ij}]} \]

\( \forall j \in N_i \)

Naïve Information
Thanks are due to:

ECLT group: Irene Poli, David Ferrari, Davide De March, Debora Slanzi, Michele Forlin, Alessandro Filisetti, Davide De Lucrezia, Giovanni Minervini and Elena Lynch.

IRIDIA (ULB – Brussels): Thomas Stützle, Mauro Birattari


References


