Data Design Nature-Inspired Computing

Project Presentation

Francesco Pelosin

Dept. Environmental Sciences, Informatics and Statistics
University of Venice “Ca’ Foscari”

December 15, 2016

Content

Introduction

In cellular biology, a **vesicle** is a small structure within a cell generated by amphiphilic molecules.

They are relevant in many technological applications such as drug delivery in medicine.

A particular set of molecule, under particular experimental condition and variables, may self-assemble and originate vesicles.

Introduction

In this project we are interest in the **vesicle formation:**

- A set of real experiments have been conducted in laboratory
- A stochastic optimization technique is then used to optimize the research process
- The optimization technique allows extraction of features inside the high dimensionality of the search space
- The optimization step permits to find best levels for the factors which influence the phenomenon

Introduction

The optimization technique is a nature inspired approach, a method that emulate nature.
In particular we used the **"Ant Colony Optimization"** algorithm (ACO).

The ant colony optimization is a probabilistic technique for solving optimization problems inspired by the behavior of real ants, in particular when they harvest food.

Problem statement

As mentioned in the introduction, to understand the problem of vesicle formation, we have to perform some experiments.

For each experiment we have to consider:

- Response (Y): Turbidity
- Composition variables and process variables (X): Reactants, PH, ion strength, temperature...
- The experimental space (search space): all possible combinations of all the factors, their levels and their interactions.

Each experiment was described by a mixture of factors $(x_1, ..., x_{16})$ where x_i represents the number of volume units of X. Each x_i can assume a value between 0 and 1 with a 0.2 step.

Problem statement

Constraints

Each wheel (combination of factor) has to satisfy a property for which the sum of the levels of the factors must equal to one, more formally:

$$\sum_{i=1}^{16} x_i = 1$$

Where x_i is the level of factor i. As an example, an experiment wheel looks like:

0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.4	0.4	**0.12937**

The goal is a combinatorial problem and consists to find the best composition between the factor levels.
The total number of points in the search space, is 15.504 and correspond to all the possible mixtures.

Problem statement

Goal

Find the optimal mixture with the minimum amount of experiments

Solution proposed

The **main steps** performed by the ACO algorithm are:

1. Ants starts by exploring the environment randomly
2. When an ant discovers a source of food, it returns back to the nest laying down pheromone on the trail
3. Other colony ants start following short paths with more probability thanks to pheromone trails
4. Shortest path becomes more and more attractive while longest disappears thanks to evaporation
5. Eventually the whole colony choose the shortest path

Implementation

How do we emulate nature?

1. The environment is modeled through graph
   - Concrete: 6×16 matrix P
2. We initialize all the matrix entry equal to 1, this ensures an initial equal probability to all the paths.
3. For each generation k ants randomly choose a valid path (sum = 1) and release some pheromones.

How do we evaluate the results?

- We perform experiments (we look inside the dataset)

Implementation

Initial state of *P*. The ant starts building the path...

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0 (0)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1 (0.2)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2 (0.4)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3 (0.6)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4 (0.8)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5 (1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

 it calculates the probabilities by looking to the pheromone of first column. In this case:

$$sum = 1 + 1 + 1 + 1 + 1 + 1 = 6$$

Implementation

Computes the probabilities to choose an entry, and then perform the choice

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0 (0)	$1/6$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1 (0.2)	$1/6$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2 (0.4)	$1/6$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3 (0.6)	$1/6$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4 (0.8)	$1/6$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5 (1)	$1/6$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Let's say that 0.4 for x_1 is selected.

Implementation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0 (0)	1	1/4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1 (0.2)	1	1/4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2 (0.4)		1/4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3 (0.6)	1	1/4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4 (0.8)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5 (1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Implementation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0 (0)	1	1	1/3	1	1	1	1	1	1	1	1	1	1	1	1	1
1 (0.2)	1		1/3	1	1	1	1	1	1	1	1	1	1	1	1	1
2 (0.4)	1	1	1/3	1	1	1	1	1	1	1	1	1	1	1	1	1
3 (0.6)	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4 (0.8)	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5 (1)	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Current sum = 0.6

Implementation

The ant will eventually find a valid path and an associated Y (experiment) we then leave some pheromone to remember the path.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0 (0)	1	1	2	2	2	2	2	2	1	2	2	2	1	2	2	2
1 (0.2)	1	2	1	1	1	1	1	1	2	1	1	1	2	1	1	1
2 (0.4)	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3 (0.6)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4 (0.8)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5 (1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Implementation

After the first step of random exploration, we let the k ants to explore the graph by looking the pheromone matrix for a certain number of generations. In particular **each ant will**:

1. Evaluate the pheromone trails in the matrix
2. Compute the probability of a given movement from the pheromone quantities
3. Evaluate the path proposed (experiment)
4. Release pheromone for the next generation by updating the matrix according to the goodness of the path found

The ants will eventually **converge to a common path** based on the probability of choosing good old trails.

Evaluation and Results

Tuning

In order to achieve good results the algorithm required **tuning of several parameters**:

- Number of generations
- Number of ants
- Evaporation factor
- Pheromone quantity for the best solution and for other solutions

What path did **we awarded?**

- The best generation path
- The best path seen so far

Evaluation and Results

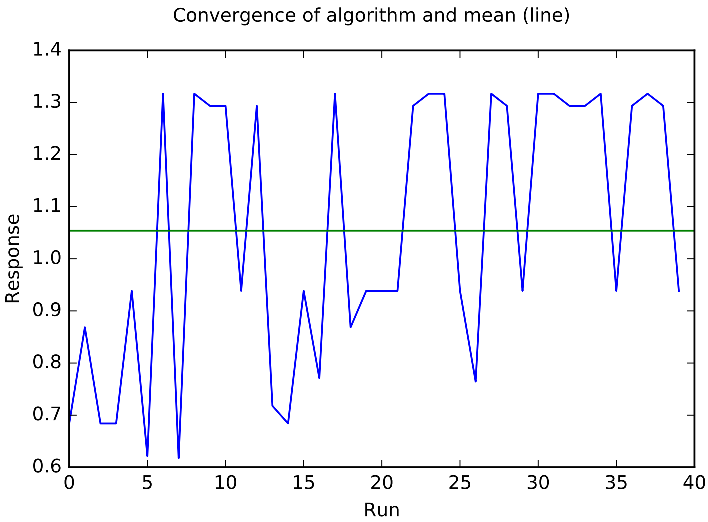
Results analysis

The algorithm achieved some **pretty good results**. We performed 40 runs to evaluate the algorithm and:

- Average best solution proposed gives a response of 1.05
- The standard deviation of the solution proposed is 0.25
- Average number of unique experiment to perform before convergence is 457, which roughly correspond to the 3% of the whole search space

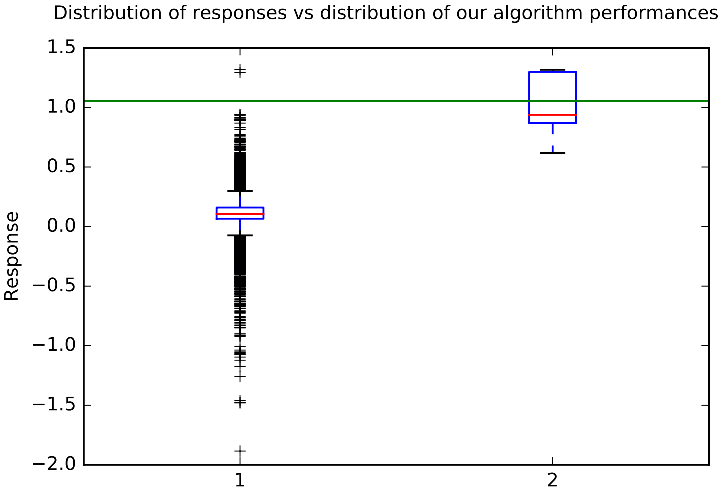
Evaluation and Results

Results analysis



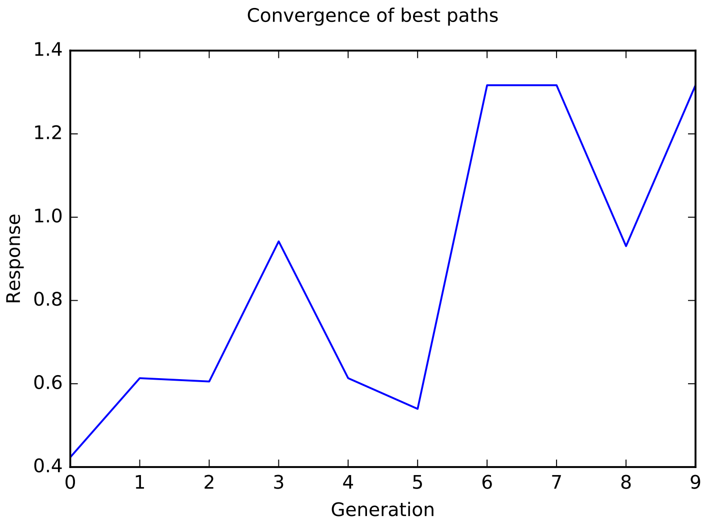
Evaluation and Results

Results analysis



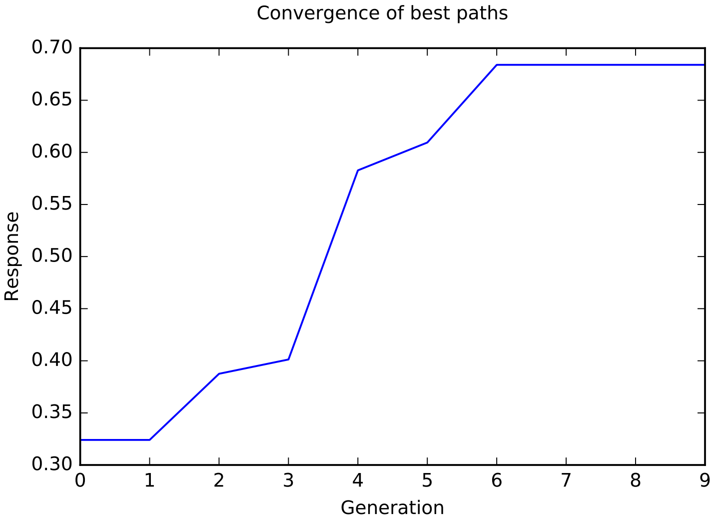
Evaluation and Results

Results analysis



Evaluation and Results

Results analysis



Thanks