Metaheuristics for Optimization

Series 1 : NK-landscape models

Return no later than September 26th, 2022 (16h00)

1 NK-landscape models

Our goal in this exercise is to illustrate a class of models known as *NK-landscape models*. These have been introduced in order to generate fitness landscapes with ruggedness that can be tuned by a single *degree* parameter K.

Therefore, let $x = (x_1, x_2, ..., x_N)$ be a binary sequence of length N, i.e. $x \in \{0,1\}^N$. The global fitness F is then given by the sum of local fitness contributions f_K :

$$F(x) = \sum_{n=1}^{N-K} f_K(x_i, ..., x_{i+K})$$

This F is the quantity we would like to optimize.

If $K = 0$, then $F(x) = \sum_{i=1}^{N} f_0(x_i)$. In such a case, F has only one maximum reached by a sequence x where the values of all variables x_i are identical.

An example

The values of f_K must be specified for all binary sequences of length $K+1$. Assume we have $K = 1$ and $N = 10$. Defining F requires specifying the values of f_1 for all chains of length 2. A possible choice would be

	00	01	10	11
f_1	-1	1	1	-1

The two chains maximizing the fitness are $x = 0101010101$ and $x = 1010101010$.

2 Hill climber methods

You are requested to use the following two optimization methods :

— *Deterministic Hill-Climbing* : Generate randomly an initial sequence of N bits. Then, among its neighbors, choose the one with the highest fitness. A "neighbor" of a given sequence x is intended to denote a sequence differing by only one bit from x. The algorithm ends when the fitness cannot be increased anymore.

— *Probabilistic Hill-Climbing* : The principle is similar to the deterministic Hill-Climbing method, except that the selection among the neighbors is performed in a stochastic manner. In particular it means that neighbors with higher fitness are more likely to be selected than

neighbors with lower fitness; their probability of selection is proportional to the value of their fitness, $P(x') = \frac{F(x')}{\sum_{y \in V(x)} F(y)}$, for $x' \in V(x)$ where $V(x)$ is the set of all the neighbors of x.Here the algorithm ends after some pre-specified number of steps (we suggest to fix this number as ten times the mean result obtained previously for deterministic climbing). The result to be returned is the *best* result reached during the exploration.
In addition, implement this method with an *aspiration* process, such that a neighbor that has a better fitness than the current best-explored solution, is always selected.

3 Work to do

You are asked to investigate how the parameter K impacts on the landscape's ruggedness for chains of length $N = 21$. Consider the three landscapes defined by the following local fitness functions f_K for $K = 0, 1, 2$:

	0	1
f_0	2	1

	00	01	10	11
f_1	2	3	2	0

	000	001	010	011	100	101	110	111
f_2	0	1	1	0	2	0	0	0

For every K, run 50 times each of the two optimization methods explained in section 2 and investigate the "stability" of obtained solutions.
Intuitively, the more rugged the landscape the less stable the solutions should be. Though different ways for assessing stability are possible, we propose to compute the pairwise Hamming distances between the 50 obtained solutions and then plot the histograms of these distances.

Describe and discuss the behaviour and the performance (in terms of the quality of the solutions) of each exploration method.

NB To calculate the Hamming distance, you count the number of bits where two same-length sequences differ.

3.1 Code

The algorithm has to take as input the parameter K and the optimization method to use, and must return the sequence x maximizing the fitness, as well as the corresponding value of the fitness.

The use of python notebooks is highly recommended, to be able to include code, results, and comments in the same file. Graphs must include a title and a legend for the axes when needed.

To start, you can take advantage of the given skeleton code.

4 Work to return

For this series, submit your code, results, and comments, and upload them on moodle, by Monday, September 26, 2022 at 16h00.

In the comments, answer the following questions :

1. What is the roughness of the landscape ?
2. What is the research space of the employed optimization methods ?
3. How are the two methods different ?
4. What is the neighborhood of an element of the research space ?
5. Is the solution of the optimization an element belonging to the research space or the fitness space ?