Stochastic optimization algorithms 2018 Home problem 2

Martin SOGNO - Personal ID Number : 19970324-T233

October 2018

Problem 2.1, 5p, The traveling salesman problem (TSP)

a)

The total number of paths in case of n cities is equal to $n!$ (it can be proven with a simple recurrence). If we consider a particular path, it is equivalent to each circular permutation of itself and of its opposite. More formally, if we consider the path $a_1, a_2, ..., a_n$, it equivalent to $a_i, a_{i+1}, ...a_n, a_1, ..., a_{i-1}$ and $a_i, a_{i-1}, ...a_1, a_n, ..., a_{i+1}$, which is $2 * n$ paths.
So finally, the amount of different paths is $\frac{n!}{2*n} = \frac{(n-1)!}{2}$.

b)

The solution is in the file GA21b.m

c)

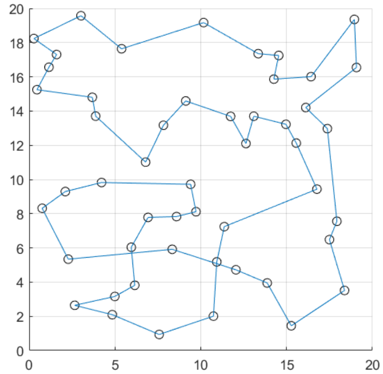
The solution is in the file AntSystem.m

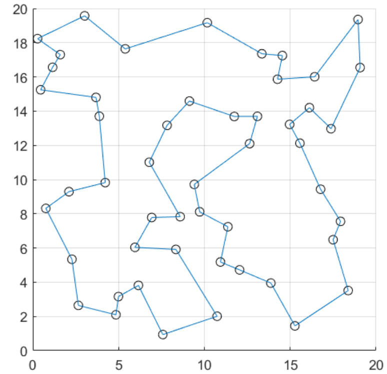
d)

The solution is in the file NNPathLengthCalculator.m The GA, with a best length of 134.02, manages to be better than the Nearest Neighbour path, which has a length of 146.39.

e)

Figure 1: Best Path with Genetic Algorithm (left) and Ant System (right)



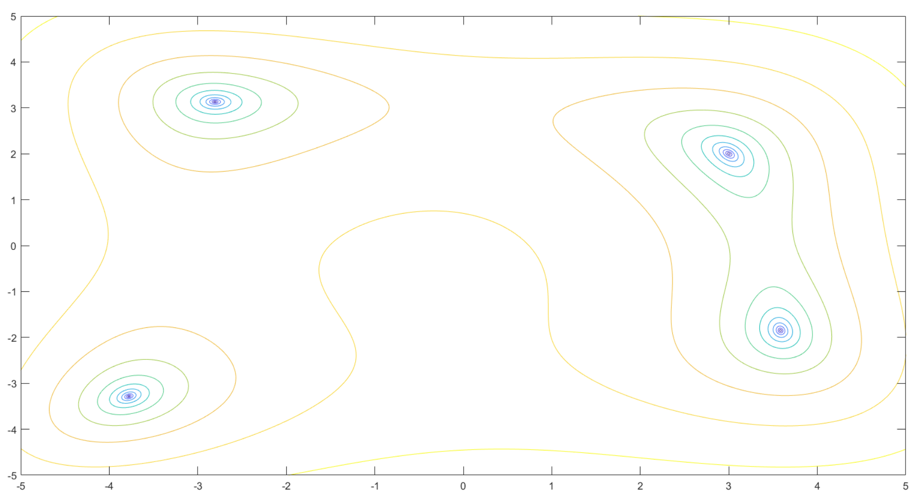


Best path with Genetic Algorithm (length : 134.02) : 18 21 24 31 33 34 39 41 44 30 29 28 20 13 8 14 17 16 19 23 26 25 12 6 3 7 22 32 36 40 48 46 47 45 42 50 49 43 37 38 35 27 15 9 1 5 4 2 10 11
Best path with Ant System (length : 121.76) : 25 26 30 29 32 36 40 48 46 47 44 41 39 42 45 50 49 43 37 38 35 27 15 9 1 5 4 2 10 11 12 6 3 7 8 13 14 17 20 28 22 16 19 23 18 21 24 31 34 33

The two paths share some parts, but are still very different, as the one obtained through genetic algorithm does not manage to avoid loops. It still manages to be a little better than the Nearest Neighbour path, which has a length of 146.39.

Problem 2.2, 2p, Particle swarm optimization

Figure 2: Contour of g



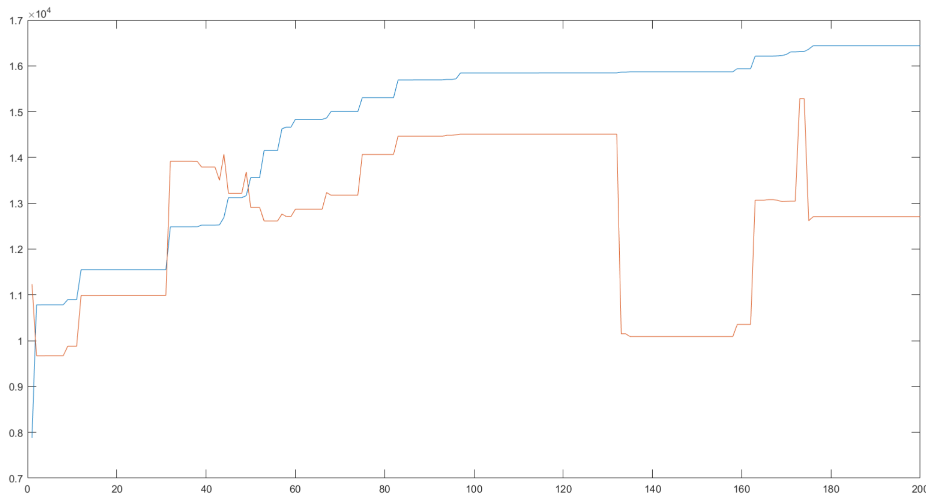
We find 4 minima, which seems to be, very approximately : (-3.7, -3.3), (3.7, -1.8), (3,2) , and (-2.9, 3.2).

x	y	$f(x,y)$
3.584428	-1.848126	0
3.000000	2.000000	0
-3.779310	-3.283185	0
-2.805118	3.131312	0

Problem 2.3, 4p, Optimization of braking systems

The truck model is implemented in EvaluateIndividual (as it is used to evaluate the efficiency of one set of weights).
The main program is called GA23.m. The test program is TestProgram.m. Here is the plot of the training and validation fitness, over the 200 generations, which seems quite classical.

Figure 3: Training (blue) and Validation (orange) fitness, over the 200 generations



Due to the low amount of training and validation data, there seems to still be some overfitting, but we manage good general results on validation and test sets. The truck manage to get to the end of the slope in 80 % of cases.

Problem 2.4, 4p, Function fitting using LGP

After several tries, I chose to reduce as much as possible the number of registers to accelerate the computations. I chose only one constant, set to 1, and 3 variables. Raising the size of the population seems useful until it reaches 500, and I chose to have 5 000 generations, because improvements were slow after this point.

The best chromosome is the following, its size is 48 genes (12 operations) :

321143312121133422214342433121211334413311324131

After the translation effectuated by the function ReadChromosome, we get an algorithm which gives a result of this form :

$$g(x) = \frac{\frac{1}{x^3-x^2}+1}{\frac{1}{x^3-x^2}+1+x} \tag{1}$$

By setting the terms to the same denominator, we get

$$g(x) = \frac{1-x^2+x^3}{1-x^2+x^4} \tag{2}$$

This function has an error under 10^{-8}, we can suppose that it is the good function and that error is only comming from round up and avoiding division by zero term.