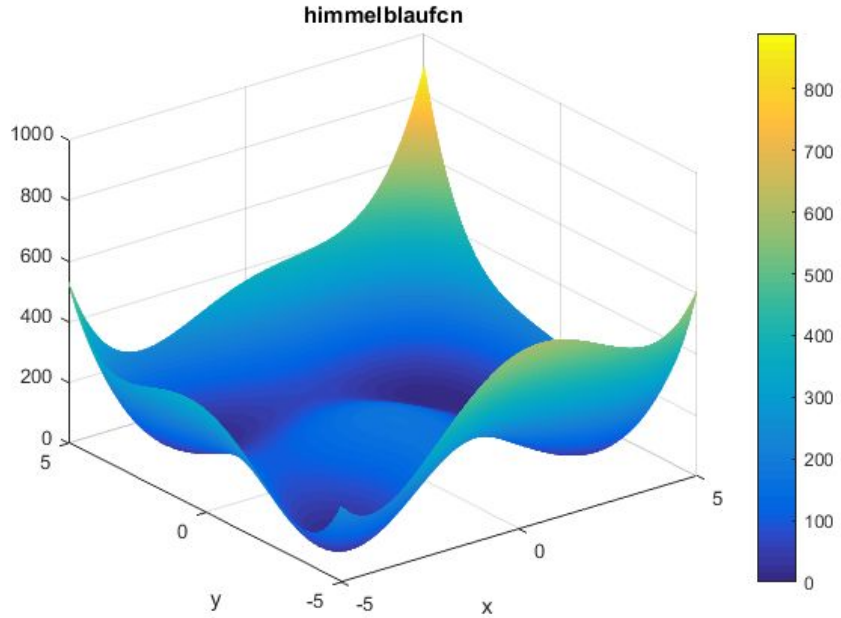
Lista VI: Gradiente Descendente e Algoritmo Genético

Vilmar Neto

Development tools

The test functions: Himmelblau's function

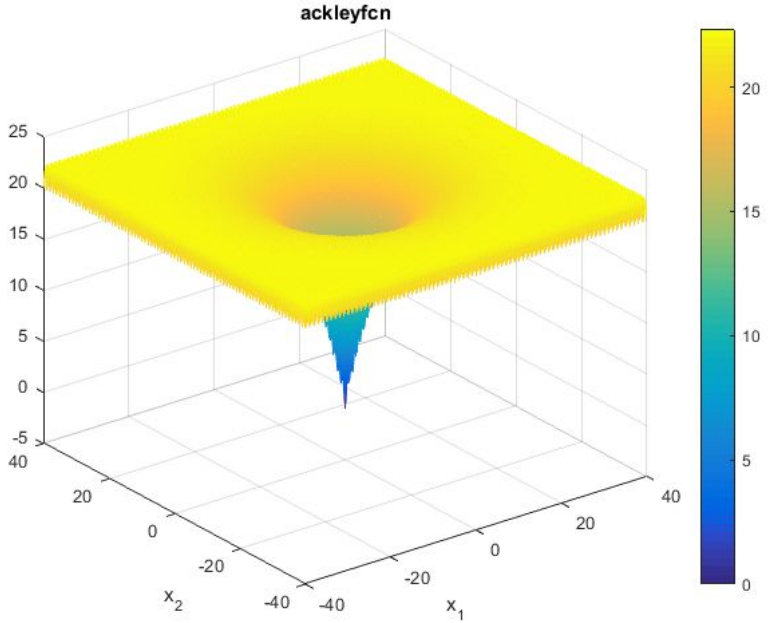


$$f(x) = (x^2+y-11)^2 + (x+y^2-7)^2$$

local minima at:

- f(3,2) = 0
- f(-2.805118,3.131312) = 0
- f(-3.779310,-3.283186) = 0
- f(3.584428,-1.848126) = 0

The test functions: Ackley's function



$f(x,y) = -200e^{(-0.2 * (x^2+y^2)^{0.5})}$

global optimum point at:

$f(0,0) = 0$

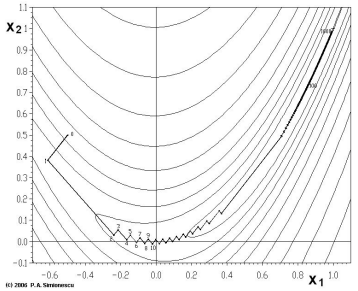
The Algorithms:Gradient Descendent

Imagine you have a function, like f(x) = x² + 4x -16. And you want to find the x that results the minimum result.

Let's call that value of θ (theta). α is a value between 10^-1 and 10^-6.

$$\theta_1 = \theta_0 - \alpha f'(\theta_0) \quad \alpha > 0$$

Iterating this process a sufficient number of times, you will encounter the theta.



Conditions for Gradient Descent to work

- The function must be convex
- Alpha cannot be too large nor too small
- Depending the function, alpha value will change
- If the function have some local minimum, it may not go to the global minimum

The Algorithms: Genetic Algorithm

1. Initiate the population size
2. Calculate the fitness & keep track of the king
3. Select the parents
4. Mutate 20% of the parents
5. Evolve and generate a new population
6. Repeat 1 to 5 until king does not change for 200 generation

GA: The Evolution

This functions receive a population and return another population.

The first step is to generate a list of pairs of each individual (pair(fitness, individual)). After that we sort the list by the fitness. This list will be sorted from the worst individual to the best.

With a sorted list we remove X% of the best individuals to be the parents. after that we include as parent another X random individuals and mutate a few.

GA: The Evolution

With the list of parents we calculate how many individuals are missing to the next pop.
number of children = (POP_SIZE - len(parents))

In a loop we just randomly chose a mother and a father (from the parent list) and mix the two individuals (half of the string from the father and half from the mother).

After that we just return the new generation to the main function.

Results: Himmelblau

Descent Gradient

	x	y	f(x,y)
Média	2,99	2,00	0,00
Desvio	0	0	0

Genetic Algorithm

	x	y	f(x,y)
Média	2,99895	2,00	0,00968
Desvio	30,43	1,22	0,0065

Results: Ackley

Descent Gradient

The result diverge because of its properties of calculation

Genetic Algorithm

	x	y	f(x,y)
Média	-0,00065	-0,00155	-199,2789
Desvio	0,00267	0,00063	1,42475

The code

https://github.com/Dkmister/BioComp2018/tree/master/Trabalho6

Conclusion

- GA is faster than Gradient Descent;
- Gradient Descent can have better approximation , depending the function and parameters;
- GA won’t go to local minima;