WSI Task nr 2: Evolutionary algorithm

Piotr Lenczewski

November 13, 2023

Abstract

The goal of this study is to design and implement an evolutionary algorithm, study the convergence of this algorithm based on f1(simpler one) and f9(more complicated one) functions from CEC2017 benchmark, for n=10 dimensionality, compare this algorithm with gradient descent algorithm from previous task.

1 Implementation of evolutionary algorithm

1.1 What is evolutionary algorithm

Evolutionary algorithms are a class of optimization algorithms inspired by the process of natural selection and evolution in biological systems. They comprise of:

- Selection: Chooses better individuals from the current population with a higher likelihood than less fit ones.
- Genetic Operators: Typically involves crossover and mutation while preserving population size (the number of mutants equals the number of parents).
- Crossover: Generates an "intermediate" point between parents, applied judiciously depending on problem characteristics.
- Mutation: Generates a point from the neighborhood of the mutated point.
- Replacement: Determines which individuals advance to the next generation.

1.2 1+1 algorithm

In this study I will use 1+1 algorithm. This algorithm is a simple optimization algorithm that belongs to the family of evolutionary algorithms. It's best suited for problems where the objective is to find the optimum of a function without knowing its analytical form. It's characteristics:

- The population size that advances to next iteration is one.
- Only one offspring is generated.
- Offspring is compared to the parent and best advances.

1.3 Gaussian distribution and 1/5 method

- Gaussian distribution gives the higher chance of drawing a number the closer to the center. In this case the center would be the parent.
- 1/5 method is method of updating deviation (the number which indicates the spread or dispersion of the distribution). It's implementation requires the chance of success or chance of offspring being better than parent to be close to 20 percent or 1/5 (chance is estimated using previous iterations).

1.4 Stop conditions

- reaching max iterations
- average improvement of last n iterations below tolerance
- last m iterations without success

1.5 Pseudocode for the 1+1 algorithm

Variables:

- f(x) - function to optimize
- x0 - starting point
- d = deviation
- b = deviation change
- a = deviation change frequency
- tol = tolerance

```
begin
    parent = x0
    successes = 0
    i = 0
    while stop conditions not met:
        child = gauss(d, parent)
        if child < parent:
            parent = child
            successes += 1
        end
        if i % a == 0:
            if successes/a > 1/5:
                d = d * (1 + b)
            else:
                d = d * (1 - b)
            end
        end
        i += 1
    end
end
```

2 Planned numerical experiments

For numerical experiments I will be using program given with this document.

2.1 Set up of not studied variables

- Objective function - f1 and f9 function from CEC2017 benchmark
- Starting point = (100, 100, ..., 100) in 10 dimension
- max iter = 500
- tolerance = 1e5
- deviation change = 0.2
- deviation change frequency = 10

2.2 Set up of stop conditions variables

I will by try and error find best values of n and m for currently set up values. For rest of studies only max iter condition will be active.

2.3 Influence of deviation variable and its derivatives on convergence of 1+1 algorithm

To study influence of deviation I will present and interpret multiple uses of the algorithm for different values of this parameter.

2.4 Comparison of 1+1 algorithm to gradient descent

For comparison of this two algorithms I will need to get an average of multiple (in this case 15) uses of 1+1 algorithm as it is based on randomness. I will also use two objective functions f1 and f9, which have different complexity levels.

3 Results of numerical experiments

3.1 Set up of stop conditions variables

The question of best n and m parameters values seems to be dependent on objective function and our goal. For example average improvement of last n = 250 iterations below tolerance seems to affect mostly f9 function. The last m = 200 iterations without success affects mostly f1 function. I've tried to use values that don't prevent some of the use cases from reaching max iterations.

3.2 Influence of deviation variable and its derivatives on convergence of 1+1 algorithm

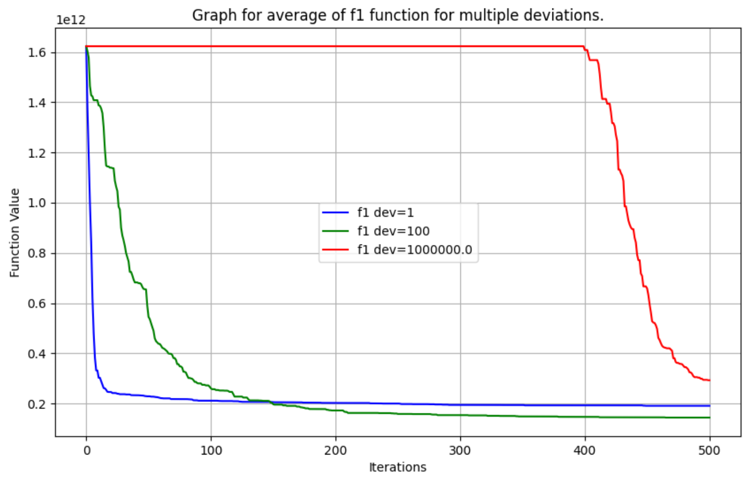

Figure 1: f1 function minimisation for different deviations

Table 1: Values found after 500 iterations for different deviations

dev = 1	173069889227
dev = 100	147924570453
dev = 1e6	281370840902

What we see on the graph is that for smaller deviation we move faster to the local optimum, but it is harder to find other local optimums. On the other hand for bigger deviation it is easier to get close to better local optimums but it is harder to get close enough to the center of optimum.

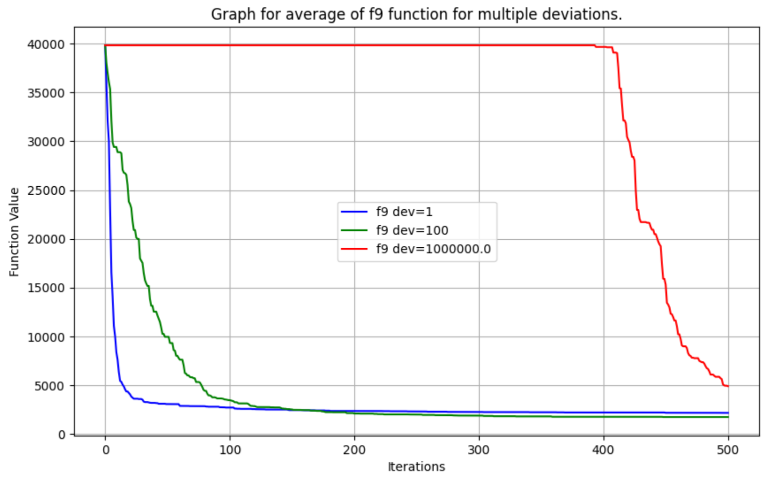

Figure 2: f9 function minimisation for different deviations

Table 2: Values found after 500 iterations for different deviations

dev = 1	2083
dev = 100	1798
dev = 1e6	4715

For f9 function the results are similar as for f1 function. From this moment I set deviation as 100.

3.3 Comparison of 1+1 algorithm to gradient descent

All tests were conducted with 500 iterations.

	1+1	gradient descent
f1	152415191457	6832
f9	1744	39728

Table 3: Achieved minimised values of function based on minimisation method

1+1 algorithm reaches better values for f9 but on the other hand gradient descent reaches better values on f1 function.

4 Conclusion

- n and m variables for stop condition should be set accordingly to objective function and goal of researcher
- deviation variable should be set somewhere in the middle, as low deviation means low exploration capabilities and high deviation means low exploitation capabilities
- Gradient descent method is better for minimising simple functions like f1. 1+1 algorithm is better suited for more complicated functions like function f9.

References

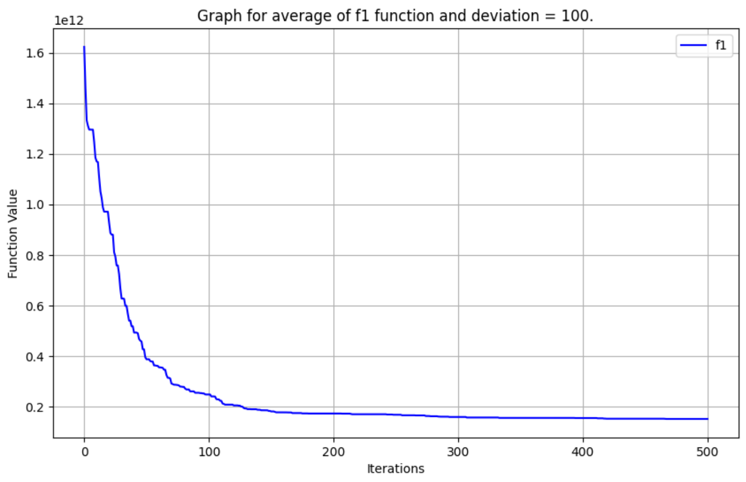

Figure 3: Average graph for f1 minimisation using 1+1

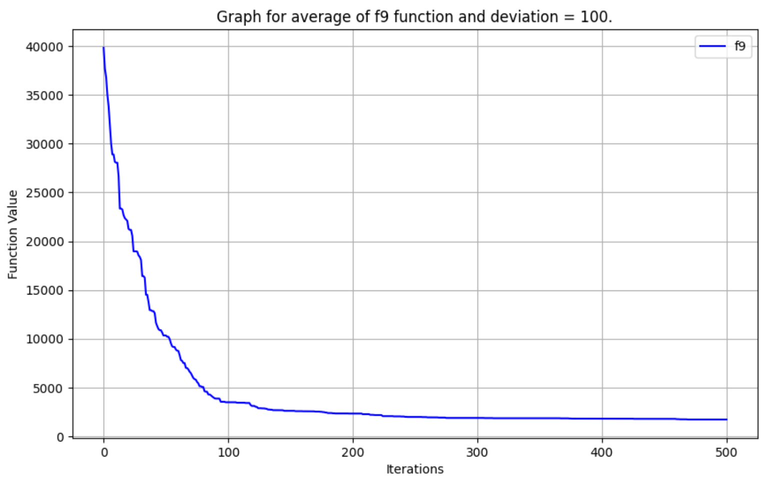

Figure 4: Average graph for f9 minimisation using 1+1

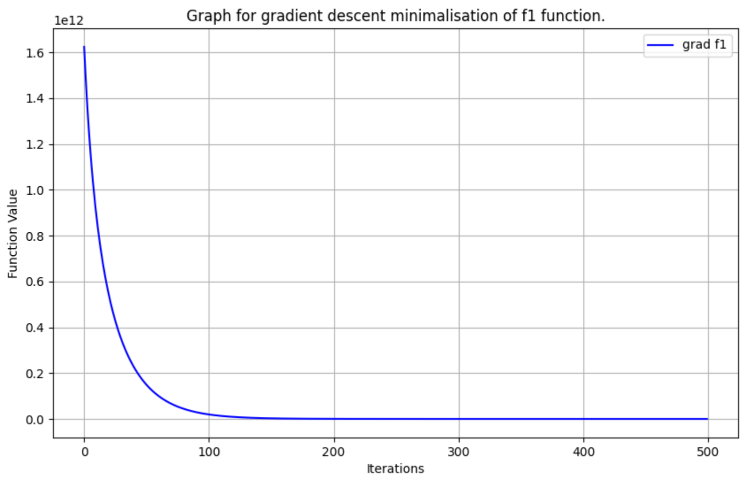

Figure 5: Graph for f1 minimisation using gradient descent

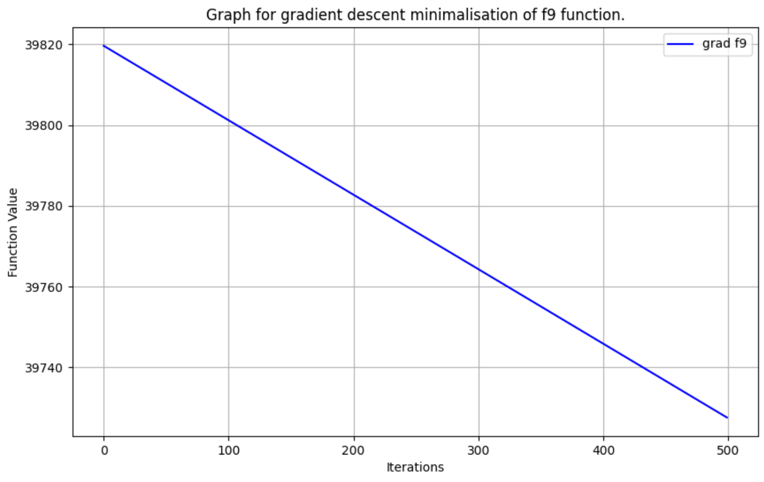

Figure 6: Graph for f9 minimisation using gradient descent