# Evolutionary Algorithms

## Report

## Evolution strategies

**Authors:**
Piotr Pawełko
Angelika Ucherek

**Laboratory dates:**
**11.04.2022**
**02.05.2022**

The task for this laboratory topic was to implement evolution algorithm to determine 3 unknown values of a function presented below. The unknow values are a, b and c. Values of input and output for 101 points are known. For the testing we choose model 5 for which points are presented on a figure 1. Next task was to test the algorithm depending on 2 different mature approaches (μ + λ), (μ, λ). The (μ + λ) strategy selects the μ best solutions from the union of parents and offspring. In contrast, in the (μ, λ) strategy the best μ offspring's a selected from λ (λ > μ) descendants to replace the parents. Last task was to check how μ, λ values affect time need for getting the right solution. All results are presented in a tables below.

Function:

$o = a * (i^2 – b * \cos(c * \pi * i))$

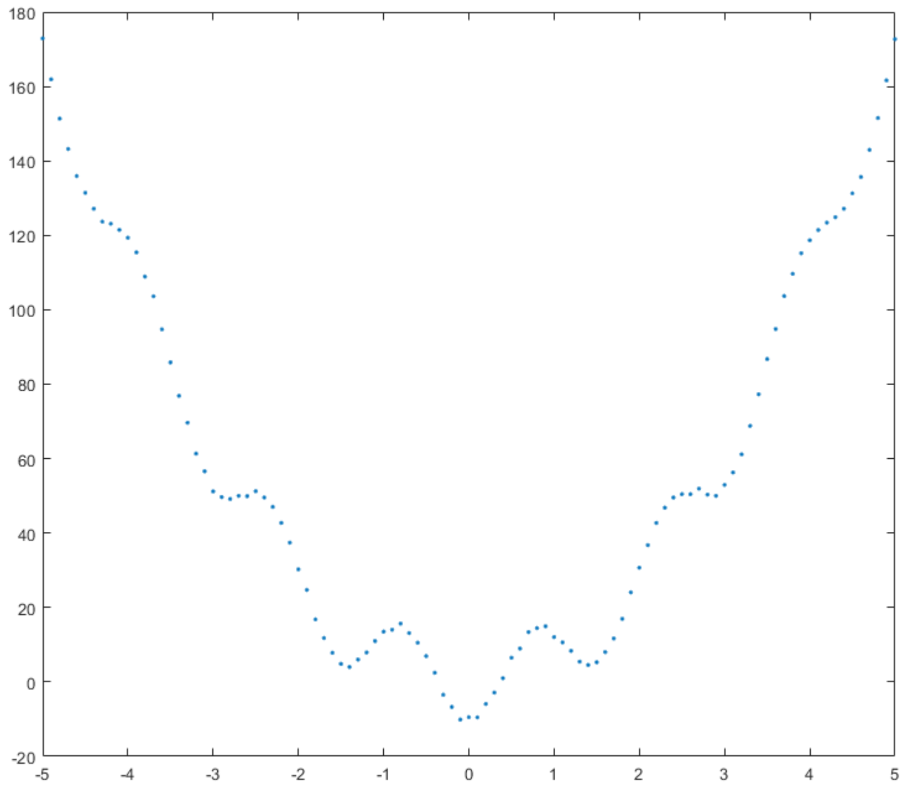


*Figure 1 Model 5*

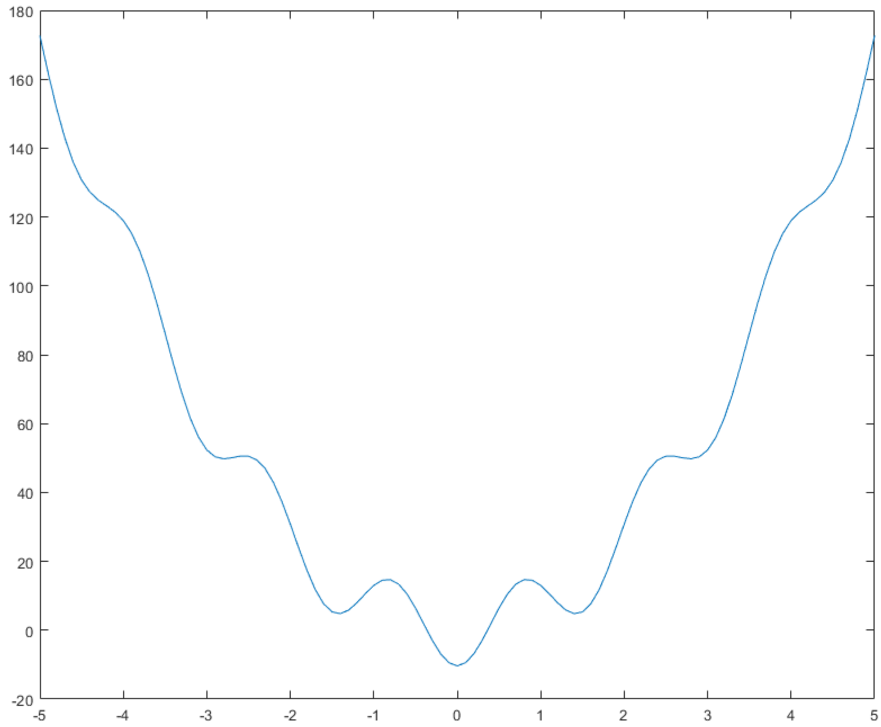


*Figure 2 Obtained function by algorithm*

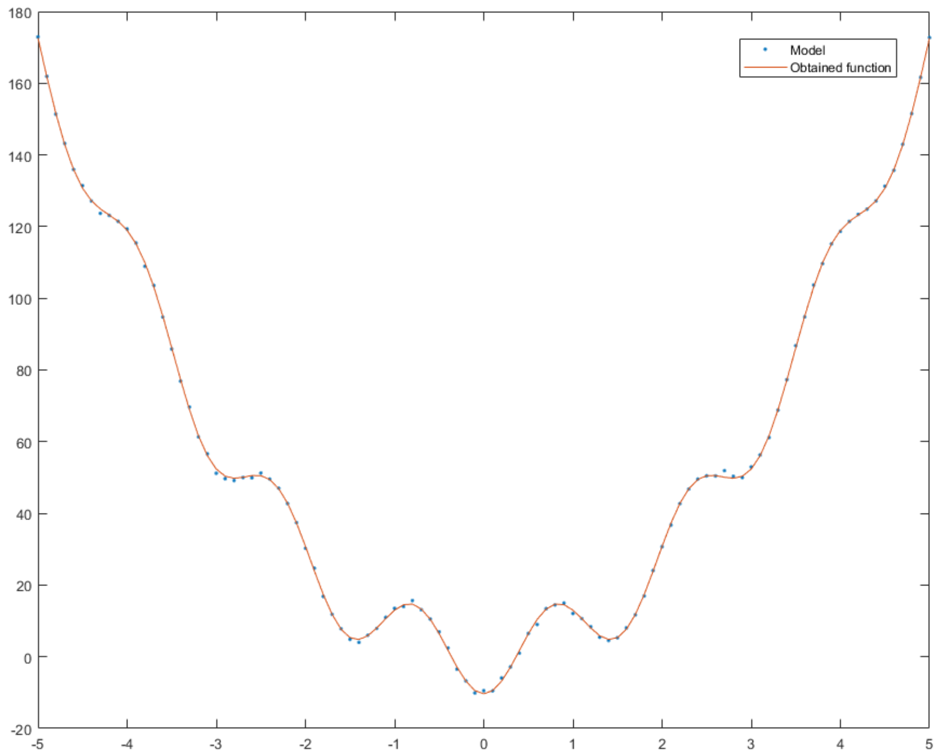


*Figure 3 Obtained function and Model*

**(μ + λ) – approach**

| λ\ μ | 100 | 250 | 500 |
|---|---|---|---|
| **5** | T = 0.539 s | T = 1.152 s | T = 2.115 s |
| **7** | T = 0.677 s | T = 1.562 s | T = 2.651 s |
| **10** | T = 0.711 s | T = 1.824 s | T = 3.274 s |

**(μ, λ) – approach**

| λ\ μ | 100 | 250 | 500 |
|---|---|---|---|
| **5** | T = 0.536 s | T = 1.226 s | T = 2.031 s |
| **7** | T = 0.638 s | T = 1.341 s | T = 2.411 s |
| **10** | T = 0.712 s | T = 1.634 s | T = 3.105 s |

**Conclusions:**

- (μ + λ) approach and (μ, λ) approach for same λ values and population sizes has nearly same results
- Increasing population has big affect on increasing the calculating time
- Increasing the parameter λ does not affect calculating that much as the increasing population does

**Code:**

```
model = readmatrix("model5.txt");


sum_err = 0;
P_size = 500;
lambda = 10;
new_pop = zeros(P_size,7);


method = 1; % 1 - lambda, 2 - lamda + u


%initialize first population
for i=1:P_size
    new_pop(i,1:3)=-10+rand(1,3)*20;
    new_pop(i,4:6)=rand(1,3)*10;
end
%evalute first population
new_pop = evaluate(new_pop, model);
```

```
epsilon = 0;
g=0;
while g < 150
    PR = new_pop(1,7);
    mutants = repmat(new_pop, lambda, 1);
    mutants = mutation(mutants);
    mutants = evaluate(mutants,model);
    switch method
        case 1
            new_pop = mutants;
            new_pop = sortrows(new_pop,7);
        case 2
            new_pop = [mutants; new_pop];
            new_pop = sortrows(new_pop,7);
    end
    new_pop = new_pop(1:P_size,:);
    PP = new_pop(1,7);
    epsilon = abs(PR - PP);
    if epsilon < 10^(-5) && epsilon > 0
       break;
    end
    g = g+1;
end

function [mutants] = mutation (Pop_to_mut)
    tau1 = 1/sqrt(12);
    tau2 = 1/sqrt(2*sqrt(6));
    m_size = length(Pop_to_mut);
    mutants = zeros(m_size,7);
    for i = 1:m_size
        mutants(i,1) = Pop_to_mut(i,1) + randn()*Pop_to_mut(i,4);
        mutants(i,2) = Pop_to_mut(i,2) + randn()*Pop_to_mut(i,5);
        mutants(i,3) = Pop_to_mut(i,3) + randn()*Pop_to_mut(i,6);
        r_sigma1 = tau1 * normrnd(0,1);
        r_sigma2= tau2 * normrnd(0,1);
        sigmaA = Pop_to_mut(i,4) * exp(r_sigma1) * exp(r_sigma2);
        r_sigma2= tau2 * normrnd(0,1);
        sigmaB = Pop_to_mut(i,5) * exp(r_sigma1) * exp(r_sigma2);
        r_sigma2= tau2 * normrnd(0,1);
        sigmaC = Pop_to_mut(i,6) * exp(r_sigma1) * exp(r_sigma2);
        mutants(i,4:6) = [sigmaA, sigmaB, sigmaC];
    end
end


function [evalueted] = evaluate(Pop_to_ev, model)
    P_size = length(Pop_to_ev);
    evalueted = Pop_to_ev;
    for l=1:P_size
        sum_err = 0;
        a = evalueted(l,1);
        b = evalueted(l,2);
        c = evalueted(l,3);
        for t=1:101
```

```
            i = model(t,1);
            o = a*(i.^2 - b*cos(c*pi*i));
            err = (o - model(t,2)).^2;
            sum_err = sum_err + err;
        end
        mid_sq_err = 1/101*sum_err;
        evalueted(l,7) = mid_sq_err;
    end
    evalueted = sortrows(evalueted,7);
end
```