EARIN Lab 3 Report

Krzysztof Rudnicki, 307585
Jakub Kliszko, 303866

April 17, 2023

1 Exercise Variant 2 - "Rastrigin function"

Our task was to write a program that optimizes Rastrigin function:

$$f(x, y) = 20 + (x^2 - 10\,cos(2x)) + (y^2 - 10\,cos(2y))$$

Using Evolutionary Strategy (μ, λ) (later refered as ES(μ, λ))

2 Implementation

Program can be ran by installing python, moving to project directory and issuing command:

```
python main.py
```

There are 7 parameters we can (but do not have to) change:

1. Number of parents (default equal to 5)
2. Size of population (default equal to 20)
3. Mutation Strength (default equal to 0.1)
4. Number of generations (default equal to 100)
5. Minimal Value (default equal to -5.12)
6. Maximal Value (default equal to 5.12)
7. Number of outputs (default equal to 10)

Number of outputs are strictly for displaying results and does not influence the result itself
To set parameters values user can add those flags to program run:

```
-nop --number_of_parents [number]
-sop --size_of_population [number]
-ms --mutation_strength [number]
-nog --number_of_generations [number]
-min --min_value [number]
-max --max_value [number]
-noo --number_of_outputs [number]
```

Order of those parameters does not matter, user can provide none, one, or any number of arguments
Exemplary use (settings all values to default values):

```
python main.py -nop 5 -sop 20 -s 0.1
-i 100 -min -5.12 -max 5.12 -noo 10
```

There are additional flags for quality of life with the program

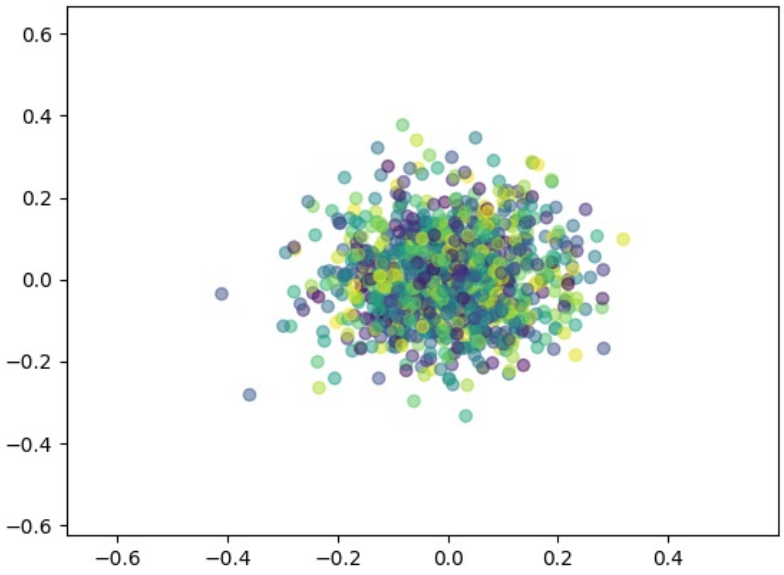
```
-nd --no_display (If used will NOT print out the plots)
-s --save (If used WILL save plot files to code folder)
```

To print help info about program user can issue help flag:

```
python main.py -h
```

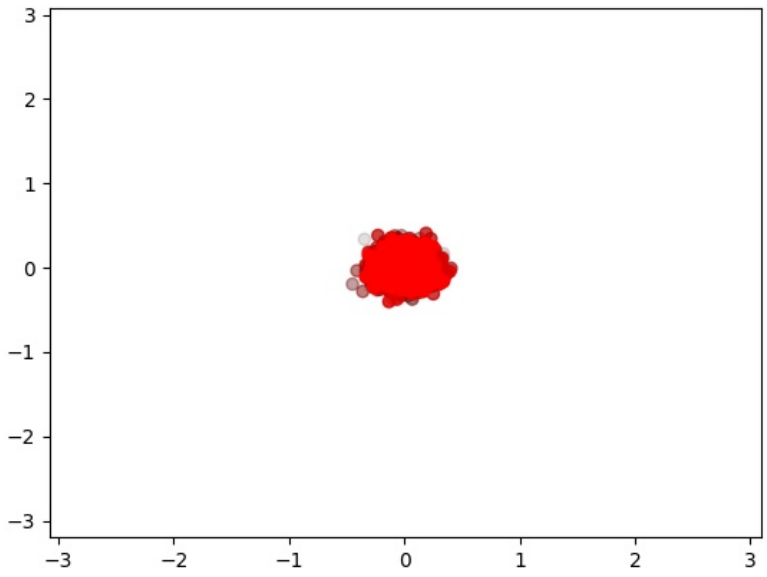
Results will be displayed on 2D scatter plot. There will be as many outputs as user wanted with incrementation of generation so that the final plot will be on final generation

Figure 1: Exemplary plot halfway through generation with parameters nop 250 sop 1000 ms 0.1 nog 500 min max (-5.12, 5.12) noo 10



At the end summary of results on plot will display with red gradient showing results from the earliest (white) to latest (bright red)

Figure 2: Exemplary summary plot with parameters
nop 250 sop 1000 ms 0.1 nog 500 min max (-5.12, 5.12) noo 10



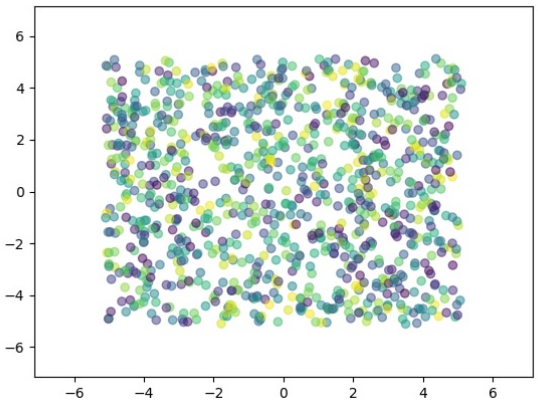
Results will be displayed and if user requested saved in the same folder as code directory for further inspection, with file name containing information about input parameters
4 more information will also be displayed:

- Best individual found - x, y coordinates closest to 0
- Best fitness found
- Total Generation Time - including ONLY generation (not plot display)
- Time per generation - Total generation time divided by number of generations

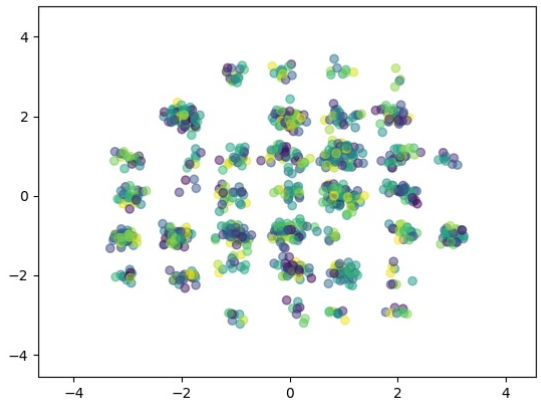
3 Results

We have successfully implemented ES(μ, λ) to optimize Rastrigin function
Rastrigin function is used to test optimization algorithms as it contains a lot of local minima, our plots should therefore tend to contain values closer and closer to zeros
Exemplary run of the program with parameters:

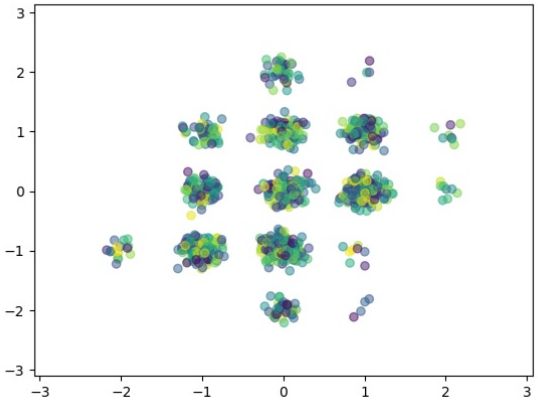
- Number of parents - 250
- Size of population - 1000
- Mutation Strength - 0.1
- Number of generations - 20
- Min value - -5.12
- Max value - 5.12
- Number of outputs - 10



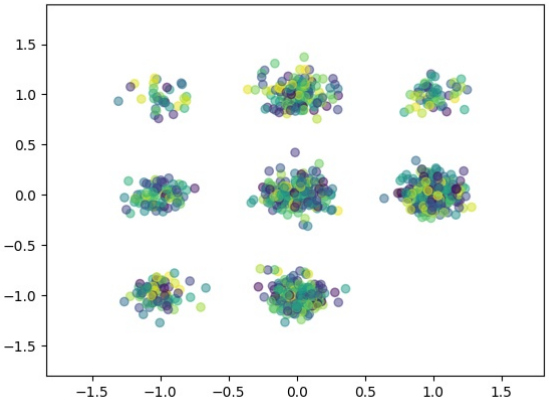
(a) 0 generation



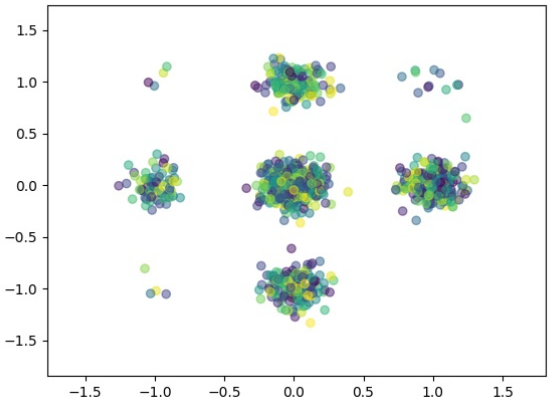
(b) 2 generation



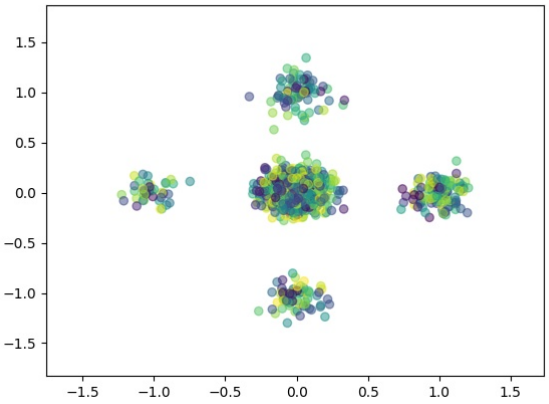
(c) 4 generation



(d) 6 generation

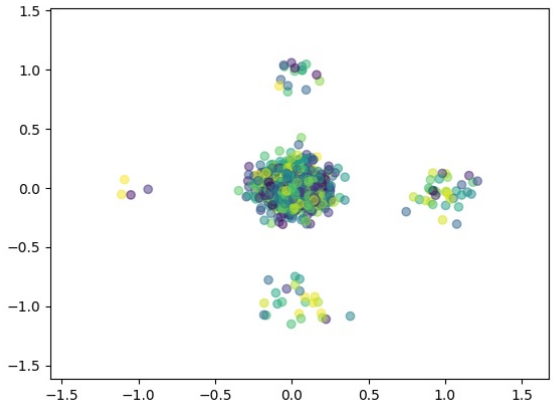


(e) 8 generation

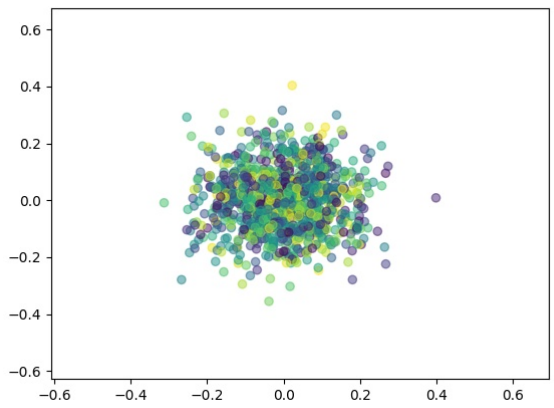


(f) 10 generation

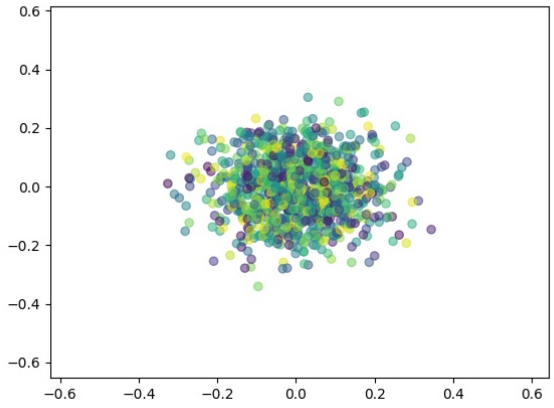
Figure 3: First 10 Generation plots with parameters nop-250:sop-1000:ms-0.1:nog-20:min-max-(-5.12, 5.12):noo-10



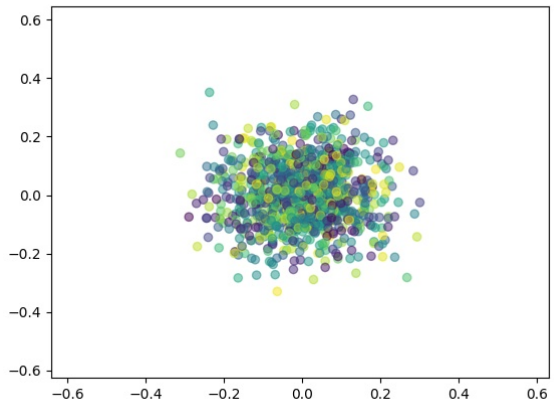
(a) 12 generation



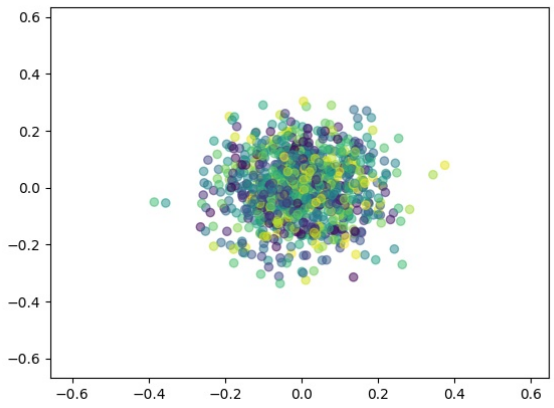
(b) 14 generation



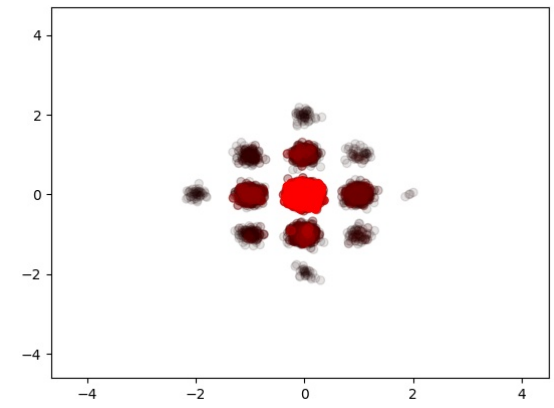
(c) 16 generation



(d) 18 generation



(e) 20 generation



(f) Summary

Figure 4: Last 10 Generation plots and summary plot with parameters nop-250:sop-1000:ms-0.1:nog-20:min-max-(-5.12, 5.12):noo-10

As we can see with every generation the result comes closer and closer to the center of the plot (point (0, 0))
At the begining (0 generation) we have random points with coordinates between -5.12 and 5.12
Which later form groups of points with center group becoming bigger and bigger (generations between 2 and 12)
To finallly reach one big group centered around (0, 0) (generations between 14 and 20)
Summary plot clearly shows that the final groups (once that most red) are close to point (0, 0)
Additonal info from execution of the program with those parameters:

- Best individual found: [-0.00512557 -0.00185114]
- Best fitness found: 0.005891436754870583
- Total generation time: 0.0532705290006561
- Time per generation: 0.0026635264500328047