

# Genetic algorithms and function minimization

Fabrice Hategekimana

## GA algorithm

Experiment GA algorithm for all combinations of the following cases:

- $p_m = 0.01$  or  $p_m = 0.1$
- no crossover or crossover with  $p_c = 0.6$  (4 cases in total)

For each of the 4 cases, run with/out keeping track of best -> (8 cases in total)

In all cases:  $N=100$ ,  $k=20$

## Statistics:

I tried to register those 4 cases.

1.  $p_m = 0.01$  & no crossover
2.  $p_m = 0.01$  &  $p_c = 0.6$
3.  $p_m = 0.1$  & no crossover
4.  $p_m = 0.1$  &  $p_c = 0.6$

I plotted the cumulative empirical probabilities (probabilities vs. fitness evaluations) for all 4 cases for:

1. optimum
2. 1% away from optimum
3. 2.5% away from optimum

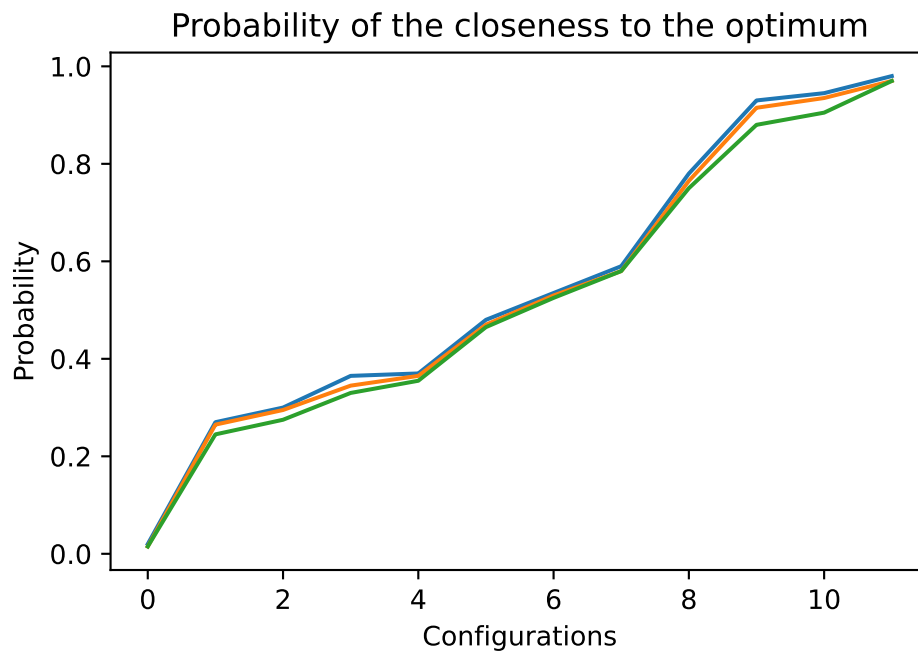
```

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("data_by_cases.csv")
df = df.sort_values(by="probs").reset_index(drop=True)
x = range(len(df))

plt.plot(x, df["probs"])
plt.plot(x, df["probs1"])
plt.plot(x, df["probs2p5"])
plt.xlabel("Configurations")
plt.ylabel("Probability")
plt.title("Probability of the closeness to the optimum")
plt.show()

```



Those are the main configurations: (p\_m, crossover, fitness evaluation).

```
df
```

```

/home/fabrice/.local/lib/python3.10/site-packages/IPython/core/formatters.py:342: FutureWarning:
return method()

```

	p_m	crossover	eval	fit_best	probs	probs1	probs2p5
0	0.01	True	10000	-1090.720693	0.020	0.015	0.015
1	0.01	False	1000	-1090.720693	0.270	0.265	0.245
2	0.01	True	1000	-1090.720693	0.300	0.295	0.275
3	0.10	False	100000	-1090.720693	0.365	0.345	0.330
4	0.10	True	100000	-1090.720693	0.370	0.365	0.355
5	0.01	False	100000	-1090.720693	0.480	0.470	0.465
6	0.10	False	1000	-1090.720693	0.535	0.530	0.525
7	0.01	True	100000	-1090.720693	0.590	0.580	0.580
8	0.10	True	1000	-1090.720693	0.780	0.765	0.750
9	0.01	False	10000	-1090.720693	0.930	0.915	0.880
10	0.10	True	10000	-1090.720693	0.945	0.935	0.905
11	0.10	False	10000	-1090.720693	0.980	0.970	0.970

I noticed that the most effective configurations are those which have  $10^3$  in the fitness evaluation. It seems that there is no need to go further to get good results.

I also notice that there is good convergence with the absence of crossover and a mutation probability of 0.10. There is also on the other hand the existence of a convergence with a crossover and a weak mutation ( $= 0.01$ ).

I deduce from this that it is rather necessary to favor one of the two parameters (crossover, mutation) and not put a lot of weight on both parameters at the same time.

#### Answer the following Questions:

1. Q: What is the Search Space of our problem?

A: The search space of our problem is the set of couple  $(x, y)$  contained in the interval  $x, y \in [10, 1000]$ .

2. Q: What are the roles of Selection, Crossover, & Mutation processes?

A: The Selection play the role of optimizing the fitness since it select the next generation according to the best fitness. But with the selection only, we dont have the power to explore for more diversity. The Crossover and Mutation processes add some variance to the set of individuals but keep in the same time the probability of the new individuals generated to stay in the subset of the best elements (from the selection step).

3. Q: What does each of the 3 processes Selection, Crossover & Mutation favor : exploration/diversification or exploitation/intensification?

A: The Selection is there for the exploitation but the Crossover and mutation are there for the diversification (exploration).