Determining the global minima of a set of functions using Genetic Algorithms

Tudose George-Daniel, 2B5

2^{nd} December 2020

1 Introduction

This paper will compare different selection and crossover methods for a genetic algorithm applied to a set of functions in trying to determine their global minimum. Genetic algorithms are an alternative to trajectory methods as are more flexible with their parameters.

2 Experimental setup

2.1 Representation

A genetic algorithm works with a collection of candidate solutions called *population*. Each candidate solution is called an *individual* or a *chromosome* and is represented using a bitstring which represents a point in the domain space. It is composed of smaller bitstrings concatenated. Each one is called a *component* and represents a point in a single dimension of the domain. Each bit of a bitstring is called a *gene*. Using this representation, a genetic algorithm works with data and not with numbers.

- Let a and b be the extremes of interval on which the function is defined and *precision* being the decimal precision.
- The length of a component is given by the formula $\log_2(10^{precision} \cdot (b-a))$. Let this length be l.
- The bitstring represents numbers in the $[0, 2^l - 1]$ interval. Let this number be B
- Let S be $\frac{B}{2^l-1}$. It represents the number in the $[0, 1]$ interval.
- By multipying with $(b-a)$ and adding a, B gets transported into the $[a, b]$ interval.

2.2 Selection methods

For this experiment, there have been used 3 selection methods.

2.2.1 Roulette Wheel

It is also knows as **Fitness proportionate selection**. It is a genetic operator which selects potentially useful solutions for recombination. It uses the fitness levels of the individuals to associate a probability of selection with each individual.[20]

Let f_i be the fitness level of the chromosome i, then the probability of selecting the i chromosome out of the **N** chromosomes is:

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i}$$

A portion of the wheel is assigned to each of the possible selections based on their fitness value.

The analogy to a roulette wheel can be envisaged by imagining a roulette wheel in which each candidate solution represents a pocket on the wheel; the size of the pockets are proportionate to the probability of selection of the solution. Selecting N chromosomes from the population is equivalent to playing N games on the roulette wheel, as each candidate is drawn independently.[20]

2.2.2 Tournament

Tournament selection is a method of selecting an individual. It involved running several "tournaments" among a few chromosomes chose at random from the population. The winner of each tournament will be slotted into the new population. In a tournament of size k, the best individual is chosen with a probability p, the second best with probability $p \cdot (1-p)$, the third best with the probability $p \cdot (1-p)^2$ and so on. Deterministic tournaments selection selects the best individual in any tournament, $p = 1$.[24]

There are two variants of the tournament selection: with replacement and without replacement. In the former, an individual who participated in a tournament can be re-elected to compete in another one. In addition, the same individual can compete with itself thus producing noise with this variant. In the latter, an individual who has been selected for a match cannot be selected again for the next match and no same individual participates in a match, so that each individual has the same opportunity to be selected for a match.[23]

The tournament selection procedure follows two steps which are repeated until the population size desired is satisfied[25]:

- a group of k individuals are chosen at random from the current population
- the best individual is selected for the mating pool

Commonly used tournament sizes are: 2,4 and 7[26]. These sizes are used to illustrate how the tournament size affects the selection behaviour. As the size of the tournament grows the selection pressure grows as well. With more individuals in a tournament, it is less likely for a less fit individual to be selected as the winner of a tournament.

For this experiment, the deterministic tournament selection with replacement has been used and the tournament size k belongs in $\{2, 7\}$. In this experiment, the selection for the tournament is done this way:

- let *popSize* be the size of population
- let r be a random integer belonging in [1,popSize]

- the r^{th} individual is selected to participate in the tournament
- repeat until the desired size of the tournament is achieved

2.2.3 Elitist

Introduced by Ken De Jong(1975), elitism is an additional scheme to any selection method. It keeps the k best individuals at each generation to be transferred into the next generation[19].

For this experiment, at any generation, the 20 best individuals of the current population are selected to perform a crossover operation resulting in an additional 20 individuals added to the current population. From the new population, a percentage p is selected to be kept in the new population. To fill the population back to its original size, a **Roulette Wheel selection** is performed on the individuals that have not been selected.

For this experiment, $p \in \{4, 10\}$.

2.3 Crossover methods

It is a probability-based crossover selection. Let p_{cx} be the general crossover probability.
For each individual a random number $r_i \in [0, 1)$ is assigned representing the crossover probability. The population is sorted based on this probability. All individuals that have a lower probability than p_{cx} are selected for the crossover operation. If the number of participants is odd, then there is a 50% chance that the last individual will perform a crossover with the next one.

2.3.1 Replacement

This crossover operation is applied on 2 individuals. Let cp be a random integer representing a single cut point around which the operation will be performed on. The crossover operation is applied on the 2 individuals and their offspring replace them in the population.

2.3.2 Elitist

The crossover operation is applied on the 2 individuals resulting in 2 offspring. Between the 4 of them, the 2 best fit individuals are kept in the population.

3 Experiment

For the experiment, a set of functions was selected for the genetic algorithm to be applied to determine the global minimum of each function. To measure the accuracy and the amount of time it takes to arrive at a result we've tested the algorithm with different number of dimensions. The precision of the values that was used is 10^{-5}. The mutation probability 1%. The crossover probability is 20%. These probabilities are constant across the experiment. The sample size is 30.

The following functions were tested with a population size equal to 100 and 200. They have also been tested with 1000 and 10000 generations. *Roulette Wheel* will be referred as **W**, *Tournament* as **T** and *Elite* as *E*. *Replacement* will be referred as **R** and *Elitist* as **E**. Let δ be the notation for the time measured in seconds and let **CX** be the notation for the crossover method. Let **G** be the notation of the number of generations. Let **D** be the number of dimensions.

The experiment has been benchmarked on an **AMD Ryzen 5 3600x**.

3.1 De Jong's function 1

It is also known as the sphere model. It is continuous, convex and unimodal.

$$f(x) = \sum_{i=1}^{n} {x_i}^2 \quad x_i \in [-5.12, 5.12]$$

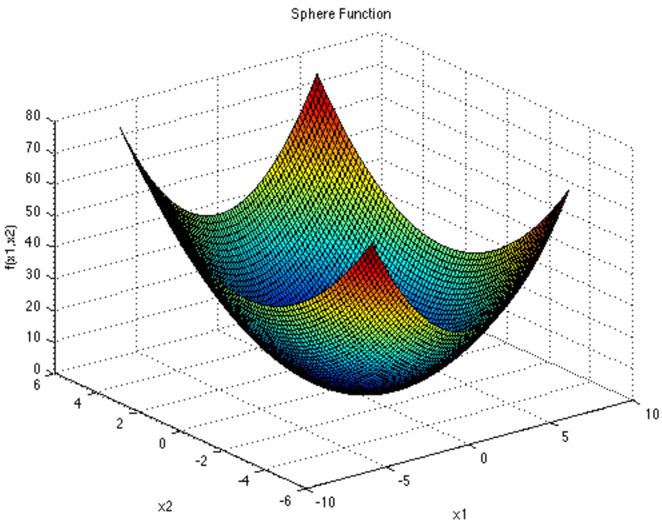

Figure 1: De Jong's function 1[7]

Dimensions	Global minimum
5	0
10	0
30	0

Table 1: De Jong's sphere function known global minima[3]

3.1.1 Fitness function

Let f_1 be De Jong's function and D_1 its fitness function.

$$D_1(x) = (f_1(x))^{-8} \quad ; \quad x \in [-5.12, 5.12]^n, \quad n \geq 1$$

De Jong's sphere function has values in $\mathbb{R}_+$ and its global minimum is 0. Because of this, the fitness function is De Jong's function raised to a negative value. This means that values that are far from 0 will be less fit while values near 0 will be more fit. Because of hardware limitation, it is extremely unlikely that 0 will be reached thus avoiding the $\frac{1}{0}$ case.

3.1.2 Results

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	0.715
5	R	10000	0	0	0	0	7.229
5	E	1000	0	0	0	0	0.718
5	E	10000	0	0	0	0	6.784
10	R	1000	0	0	0	0	1.408
10	R	10000	0	0	0	0	14.162
10	E	1000	0	0	0	0	1.397
10	E	10000	0	0	0	0	12.948
30	R	1000	6e-5	6.5e-4	2.1e-4	1.2e-4	4.178
30	R	10000	6e-5	2.2e-3	3e-4	3.8e-4	39.264
30	E	1000	5e-5	4e-4	1.68e-3	1.7e-4	4.182
30	E	10000	1e-5	3.6e-4	1.5e-4	8e-5	38.49

Table 2: De Jong's function. Population 100. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	1304
5	R	10000	0	0	0	0	13.561
5	E	1000	0	0	0	0	1.335
5	E	10000	0	0	0	0	13.949
10	R	1000	0	0	0	0	2.543
10	R	10000	0	0	0	0	27.246
10	E	1000	0	0	0	0	2.547
10	E	10000	0	0	0	0	27.673
30	R	1000	0	3e-5	1e-5	0	7.69
30	R	10000	0	3e-5	1e-5	0	79.694
30	E	1000	0	1e-5	0	0	7.514
30	E	10000	0	1e-5	0	0	83.160

Table 3: De Jong's function. Population 200. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	e-5	0	0	0.978
5	R	10000	0	3e-5	0	0	9.42
5	E	1000	0	0	0	0	0.98
5	E	10000	0	0	0	0	9.487
10	R	1000	1.685e-2	9.122e-2	4.065e-2	1.771e-2	1.863
10	R	10000	1.118e-2	9.047e-2	4.083e-2	2.027e-2	18.308
10	E	1000	7.9e-4	1.06e-2	4.29e-3	2.18e-3	1.84
10	E	10000	1.22e-3	2.697e-2	5.54e-3	4.7e-3	18.443
30	R	1000	14.5845	30.4525	21.68601	4.06314	5.4431
30	R	10000	0	0	0	0	0
30	E	1000	7.91	21.1193	14.79087	3.41132	5.502
30	E	10000	8.506	19.907	13.97039	3.10695	54.534

Table 4: De Jong's function. Population 100. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	1.917
		10000	0	1e-5	0	0	18.852
	E	1000	0	0	0	0	1.909
		10000	0	0	0	0	18.721
10	R	1000	3.14e-3	4.099e-2	2.104e-2	7.89e-3	3.665
		10000	5.56e-3	3.84e-2	1.772e-2	6.99e-3	36.595
	E	1000	01.14e-3	6.47e-3	2.53e-3	1.18e-3	3.644
		10000	8.1e-4	5.87e-3	2.35e-3	1.28e-3	37.014
30	R	1000	11.879	26.4213	18.20133	3.25344	10.884
		10000	11.0076	28.6056	18.6056	3.78328	108.9419
	E	1000	6.736	14.531	10.32632	1.89222	10.6424
		10000	5.93929	16.4262	11.04394	2.42318	109.650

Table 5: De Jong's function. Population 200. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	1.021
		10000	0	0	0	0	10.255
	E	1000	0	0	0	0	1.073
		10000	0	0	0	0	10.192
10	R	1000	0	0	0	0	1.928
		10000	0	0	0	0	19.137
	E	1000	0	0	0	0	1.913
		10000	0	0	0	0	19.4878
30	R	1000	0.12245	0.40119	0.23063	6.4103e-2	5.444
		10000	0.10284	0.31039	0.21611	4.649e-2	55.476
	E	1000	7.638e-2	0.18912	0.12073	0.02856	5.535
		10000	8.491e-2	0.2012	0.13241	3.041e-2	55.590

Table 6: De Jong's function. Population 100. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	2.081
		10000	0	0	0	0	20.719
	E	1000	0	0	0	0	2.044
		10000	0	0	0	0	20.805
10	R	1000	0	0	0	0	3.9128
		10000	0	0	0	0	39.006
	E	1000	0	0	0	0	3.852
		10000	0	0	0	0	39.0868
30	R	1000	7.725e-2	0.25618	0.15159	3.840e-2	11.037
		10000	0.10011	0.22638	0.15183	3.572e-2	111.934
	E	1000	3.631e-2	0.10897	7.446e-2	1.982e-2	11.2769
		10000	3.329e-2	0.13918	7.039e-2	2.582e-2	111.702

Table 7: De Jong's function. Population 200. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	0.963
		10000	0	0	0	0	9.386
	E	1000	0	0	0	0	0.9164
		10000	0	0	0	0	9.4124
10	R	1000	0	0	0	0	1.784
		10000	0	0	0	0	18.292
	E	1000	0	0	0	0	1.830
		10000	0	0	0	0	18.426
30	R	1000	0.11496	0.31613	0.19630	3.875e-2	5.365
		10000	0.14341	0.25242	0.19137	3.433e-2	55.384
	E	1000	4.772e-2	0.13081	9.056e-2	1.898e-2	5.444
		10000	4.799e-2	0.12531	8.772e-2	1.819e-2	55.251

Table 8: De Jong's function. Population 100. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	1.8865
		10000	0	0	0	0	18.996
	E	1000	0	0	0	0	1.860
		10000	0	0	0	0	18.9344
10	R	1000	0	0	0	0	3.544
		10000	0	0	0	0	37.056
	E	1000	0	0	0	0	3.550
		10000	0	0	0	0	37.350
30	R	1000	3.547e-2	0.19841	9.702e-2	9.702e-2	3.12e-2
		10000	5.414e-2	0.16107	8.289e-2	2.112e-2	112.264
	E	1000	2.614e-2	5.330e-2	3.737e-2	5.64e-3	11.006
		10000	2.492e-2	4.341e-2	3.447e-2	5.13e-3	111.152

Table 9: De Jong's function. Population 200. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	0.916
		10000	0	0	0	0	9.495
	E	1000	0	0	0	0	0.926
		10000	0	0	0	0	9.347
10	R	1000	0	3e-5	0	0	1.810
		10000	0	0	0	0	18.506
	E	1000	0	0	0	0	1.863
		10000	0	0	0	0	18.737
30	R	1000	1.11611	2.19555	1.59099	0.26148	5.409
		10000	1.10619	2.22242	1.67441	0.28846	55.320
	E	1000	0.55377	0.97561	0.80212	0.10142	5.435
		10000	0.60634	1.03441	0.78544	0.12994	55.728

Table 10: De Jong's function. Population 100. Elitist 10% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	0	0	0	1.861
		10000	0	0	0	0	18.949
	E	1000	0	0	0	0	1.8875
		10000	0	0	0	0	19.051
10	R	1000	0	2e-5	0	0	3.649
		10000	0	0	0	0	36.8938
	E	1000	0	0	0	0	3.672
		10000	0	0	0	0	37.94
30	R	1000	0.66148	1.30247	0.97214	0.15807	11.079
		10000	0.58611	1.12637	0.84575	0.14369	111.360
	E	1000	0.29507	0.61002	0.411524	6.664e-2	10.894
		10000	0.28803	0.46076	0.36483	4.212e-2	111.283

Table 11: De Jong's function. Population 200. Elitist 10% Selection.

3.2 Schwefel's function

Schwefel's function is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction.

$$f(x) = \sum_{i=1}^{n} -x_i \cdot sin\left(\sqrt{|x_i|}\right) \qquad x_i \in [-500, 500]$$

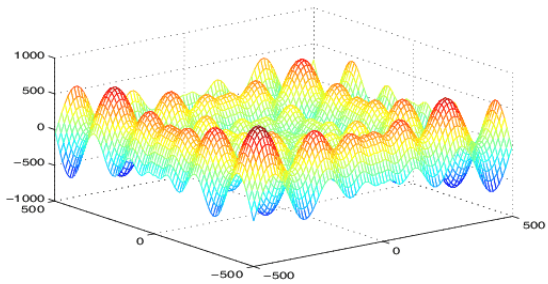

Figure 2: Schwefel's function[8]

Dimensions	Global minima
5	-2094.9145
10	-4189.829
30	-12569.487

Table 12: Schwefel's function known global minima[4]

3.2.1 Fitness function

Let f_2 be Schwefel's function and S_7 its fitness function. For this paper two fitness functions have been used. Let these functions be S_7' and S_7''.

$$\begin{cases} S_7'(x) = \left(\frac{arctan(f_2(x))+\frac{\pi}{2}}{\pi}\right)^{-25} \\ S_7''(x) = 1.015^{\sqrt{\left(\frac{arctan(f_2(x))+\frac{\pi}{2}}{\pi}\right)^{-2}}} \end{cases} , x \in [-500, 500]^n, \quad n \geq 1$$

Because Schwefel's function has values in $\mathbb{R}$, it was necessary to find a function that is defined on $\mathbb{R}$ and has a restrictive codomain. The found function was arctan as its domain is $\mathbb{R}$ and codomain is $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. It is known that for this from of Schwefel's function, the global minimum is a negative value and by applying the arctan function the function will get mapped to arctan's codomain in such a way that the minimum will be close to $-\frac{\pi}{2}$ and positive values will be above 0. The fitness function is a positive function so $\frac{\pi}{2}$ was added and then it was divided by π to get a restrictive interval to raise the selection pressure. Now the global minimum is a value close to 0. To get high values of fitness for a function with values belonging to $(0, 1)$, the result is raised to a negative power. The exponent was found empirically. This is the S_7' fitness function.

Through multiple experiments it was observed that this fitness function doesn't have a high enough selection pressure. Because of this, there was used an exponential function for which the exponent was initially S_7' but due to hardware limitations it couldn't be tested. While S_7'' is mathematically equivalent to

$$1.015^{\left(\frac{arctan(f_2(x)+\frac{\pi}{2}}{\pi}\right)^{-1}}$$

due to hardware limitation, S_7'' yielded better results than the function above. The base of the exponential function was found empirically.

Because S_7' didn't have great results, its data was not included in this paper but it is still mentioned as it is the base of the fitness function that was used.

3.2.2 Results

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.8	-2059.85	-2093.03166	6.27556	1.328
		10000	-2094.78	-2093.48	-2094.4353	0.29200	9.604
	E	1000	-2094.88	-2094.21	-2094.58266	0.16606	1.345
		10000	-2094.79	-2094.17	-2094.586	0.16397	9.50606
10	R	1000	-4188.87	-4153.2	-4186.06233	6.27485	2.580
		10000	-4189.17	-4186.05	-4187.70166	0.90815	18.795
	E	1000	-4189.41	-4069.64	-4184.077	21.62934	2.694
		10000	-4189.38	-4187.02	-4188.54866	0.61963	18.90926
30	R	1000	-12463.1	-11840.4	-12241.81333	148.76857	7.929
		10000	-12551.7	-12500.4	-12535.113	9.349	56.81526
	E	1000	-12458.1	-11894.3	-12307.58666	124.160	7.956
		10000	-12554.8	-12522.3	-12540.94333	6.65772	56.561

Table 13: Schwefel's function. Population 100. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.87	-2094.02	-2094.45733	0.18902	1.904
		10000	-2094.89	-2094.34	-2094.70266	0.15242	19.022
	E	1000	-2094.89	-2094.37	-2094.591	0.14905	1.859
		10000	-2094.9	-2094.2	-2094.6493	0.15065	19.249
10	R	1000	-4188.68	-4186.53	-4187.93	0.62240	3.811
		10000	-4189.53	-4187.24	-4188.39533	0.66385	37.94773
	E	1000	-4189.43	-4187.56	-4188.67633	0.43201	3.744
		10000	-4189.43	-4188.22	-4188.75866	0.31439	38.070
30	R	1000	-12501.6	-12102.5	-12321.23666	93.06507	11.206
		10000	-12556.8	-12535.7	-12548.61666	5.05958	113.375
	E	1000	-12520	-11986.4	-12337.64666	107.303	11.324
		10000	-12560.2	-12537.4	-12551.22	5.30714	113.7873

Table 14: Schwefel's function. Population 200. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.29	-2094.64433	0.13348	0.977
		10000	-2094.91	-2094.4	-2094.65166	0.1573	9.724
	E	1000	-2094.91	-2060.57	-2093.5166	6.22422	0.964
		10000	-2094.91	-2094.37	-2094.67366	0.11885	9.680
10	R	1000	-4187.15	-4019.22	-4156.124	35.68590	1.903
		10000	-4183.28	-4142.39	-4169.15233	8.75971	19.222
	E	1000	-4188.56	-4110.15	-4175.50266	20.83982	1.894
		10000	-4187.68	-4177.16	-4184.175	2.813	19.149
30	R	1000	-10035	-8495.86	-9177.82133	395.63129	5.717
		10000	-10289.6	-8834.29	-9383.55433	310.95300	57.129
	E	1000	-10848.1	-9073.86	-9962.037	373.61167	5.705
		10000	-11042.2	-9149.34	-10062.935	407.690	57.044

Table 15: Schwefel's function. Population 100. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.46	-2094.74633	0.14172	1.947
		10000	-2094.91	-2094.56	-2094.74733	0.090665	19.354
	E	1000	-2094.91	-2094.49	-2094.7356	0.11681	1.941
		10000	-2094.81	-2094.39	-2094.70033	0.09901	19.628
10	R	1000	-4184.86	-4148.65	-4173.143	7.68670	3.792
		10000	-4184.5	-4163.56	-4175.577	5.18554	38.337
	E	1000	-4189.09	-4183.91	-4187.076	1.32482	3.816
		10000	-4189.52	-4185.67	-4187.685	1.0396	38.4745
30	R	1000	-10452.8	-9233.94	-9855.716	300.94153	11.386
		10000	-11370.8	-9391.27	-10035.93	418.56131	114.702
	E	1000	-11210.1	-9967.85	-10624.11	300.93115	11.393
		10000	-11222.2	-10042.8	-10534.21666	298.60231	114.55746

Table 16: Schwefel's function. Population 200. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.81	-2094.5	-2094.673	0.08009	1.04896
		10000	-2094.81	-2094.5	-2094.637	0.08867	10.312
	E	1000	-2094.91	-2094.4	-2094.614	0.11037	1.048
		10000	-2094.81	-2094.5	-2094.62366	0.10736	10.504
10	R	1000	-4189.72	-4070.75	-4184.23233	22.31005	1.982
		10000	-4189.62	-4188.89	-4189.257	0.20463	19.750
	E	1000	-4189.52	-4154.84	-4187.01866	8.7167	1.955
		10000	-4189.62	-4188.89	-4189.267	0.17620	19.940
30	R	1000	-12142.9	-11401.9	-11815.1	202.49877	5.723
		10000	-12506.6	-12385.6	-12448.87666	32.83440	57.458
	E	1000	-12259.7	-11515.1	-11904.90666	236.70704	5.728
		10000	-12522.3	-12441.5	-12489.48	21.77849	57.556

Table 17: Schwefel's function. Population 100. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.4	-2094.67166	0.12830	2.069
		10000	-2094.91	-2094.4	-2094.67333	0.11336	21.062
	E	1000	-2094.91	-2094.5	-2094.682	0.11801	2.097
		10000	-2094.81	-2094.4	-2094.63766	0.10711	21.072
10	R	1000	-4189.62	-4188.79	-4189.27366	0.204475	3.954
		10000	-4189.52	-4189	-4189.28533	0.12266	39.764
	E	1000	-4189.62	-4189	-4189.28766	0.16424	3.962
		10000	-4189.62	-4189.1	-4189.29633	0.14135	39.686
30	R	1000	-12383.5	-11862.8	-12202.22666	131.349	11.504
		10000	-12512.2	-12440.4	-12479.49333	11.72006	115.421
	E	1000	-12443.2	-12026.4	-12218.49666	125.69742	11.539
		10000	-12530.7	-12494.1	-12511.123333	9.42742	115.848

Table 18: Schwefel's function. Population 200. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.49	-2094.733	0.09494	1.214
		10000	-2094.91	-2094.49	-2094.707	0.1221	12.468
	E	1000	-2094.91	-2094.5	-2094.70633	0.10216	1.2111
		10000	-2094.91	-2094.5	-2094.70933	0.14205	12.452
10	R	1000	-4189.61	-4120.11	0-4175.379	21.11138	2.485
		10000	-4189.59	-4188.33	-4188.84633	0.29463	24.768
	E	1000	-4189.45	-4146.51	-4183.227	13.54196	2.463
		10000	-4189.56	-4188.65	-4189.173	0.21719	25.137
30	R	1000	-11355.2	-9806.81	-10637.4733	387.789	7.369
		10000	-12336.5	-11686	-12045.44666	190.54121	74.592
	E	1000	-11637.6	-10223.9	-10970.48666	334.92574	7.259
		10000	-12487.3	-11908.2	-12314.55	131.42920	74.581

Table 19: Schwefel's function. Population 100. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.5	-2094.72066	0.09310	2.447
		10000	-2094.91	-2094.6	-2094.77466	0.10941	25.124
	E	1000	-2094.91	-2094.5	-2094.72433	0.105460	2.43213
		10000	-2094.91	-2094.5	-2094.75533	0.10318	25.257
10	R	1000	-4189.76	-4155.03	-4187.99833	6.23309	4.867
		10000	-4189.76	-4188.66	-4189.20766	0.242468	50.476
	E	1000	-4189.57	-4188.88	-4189.271	0.17786	5.052
		10000	-4189.66	-4188.8	-4189.25833	0.20525	50.131
30	R	1000	-11341.1	-9575.88	-10574.42966	407.55289	14.797
		10000	-12297	-11218	-11944.963333	238.08588	150.735
	E	1000	-11331.2	-9631.03	-10681.253	394.62144	14.825
		10000	-12461	-11700.2	-12105.84666	207.18577	151.620

Table 20: Schwefel's function. Population 200. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.49	-2094.72	0.12884	1.269
		10000	-2094.91	-2094.49	-2094.72166	0.10728	12.788
	E	1000	-2094.91	-2094.39	-2094.647	0.13610	1.233
		10000	-2094.91	-2094.5	-2094.64333	0.11090	12.312
10	R	1000	-4189.09	-4012.95	-4161.80166	43.32211	2.422
		10000	-4189.31	-4187.36	-4188.40733	0.54486	24.670
	E	1000	-4189.55	-4120.35	-4185.621	13.81011	2.455
		10000	-4189.65	-4188.42	-4189.10366	0.27577	24.976
30	R	1000	-11168.7	-9486.87	-10250.788	411.99165	7.441
		10000	-12123.4	-11075.1	-11676.3533	276.783675	73.775
	E	1000	-11687.2	-9521.37	-10502.56366	499.54363	7.398
		10000	-12157.8	-11255.4	-11796.76	229.78904	74.320

Table 21: Schwefel's function. Population 100. Elitist 10% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-2094.91	-2094.6	-2094.766	0.10457	2.484
		10000	-2094.91	-2094.6	-2094.80866	0.09365	25.291
	E	1000	-2094.91	-2094.5	-2094.73166	0.09255	2.411
		10000	-2094.91	-2094.6	-2094.77566	0.09038	25.167233
10	R	1000	-4189.41	-4154.66	-4187.48166	6.23637	4.746
		10000	-4189.48	-4188.08	-4188.69666	0.32958	50.194
	E	1000	-4189.66	-4188.74	-4189.19033	0.26827	4.846
		10000	-4189.63	-4188.56	-4189.18266	0.25392	49.149
30	R	1000	-11228.4	-9791.59	-10562.40133	379.48162	14.962
		10000	-12138.8	-10957.6	-11562.3666	263.68360	0150.128
	E	1000	-11308.6	-10026.1	-10705.14333	321.08802	14.835
		10000	-12050.8	-11172.3	-11172.3	250.34544	146.582

Table 22: Schwefel's function. Population 200. Elitist 10% Selection.

3.3 Rastrigin's function

This function has many local minima thus this test function is multimodal. The location of the minima are regularly distributed.

$$f(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot cos(2 \cdot \pi \cdot x_i)\right) \qquad x_i \in [-5.12, 5.12]$$

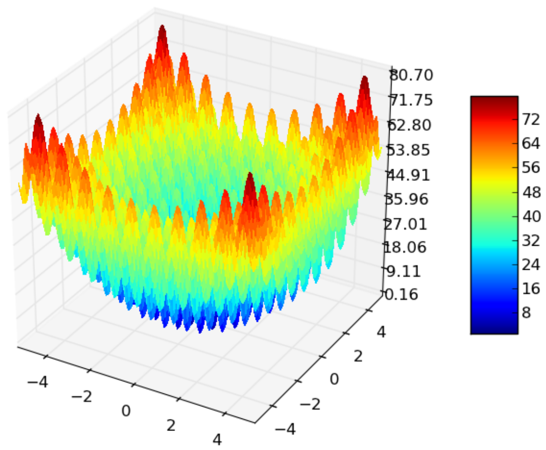

Figure 3: Rastrigins's function[9]

Dimensions	Global Minimum
5	0
10	0
30	0

Table 23: Rastrigin's function known global minima[5]

3.3.1 Fitness function

Let f_6 be Rastrigin's function and R_6 its fitness function.

$$R_6(x) = (f_6(x))^{-15} \ ; \ x \in [-5.12, 5.12]^n \ ; \ n \geq 1$$

Similar to De Jong's function, Rastrigin's function only has positive values and its global minimum is 0. Because of this a similar strategy was applied as the concepts applied for De Jong's function also apply to this one. The absolute value of the exponent was raised in trying to raise the selection pressure.

3.3.2 Results

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	12.3714	3.13256	2.76663	0.716
		10000	0	2.4834	0.78437	0.889900	7.201
	E	1000	0	6.16358	1.27744	1.39421	0.721
		10000	0	2.47297	0.94794	0.77404	7.135
10	R	1000	1.24144	19.9523	6.254917	4.38606	1.420
		10000	0	6.18413	2.455547	1.51920	14.1819
	E	1000	0	12.3958	4.55714	3.43295	1.397
		10000	0	8.66267	2.23845	1.735	14.125
30	R	1000	24.7944	49.3061	34.76536	7.10413	4.209
		10000	1.60464	26.6147	12.745215	6.47888	42.165
	E	1000	13.5127	53.0354	31.74513	9.8899	4.16386
		10000	4.35842	34.1293	15.80399	7.24689	42.207

Table 24: Rastrigin's function. Population 100. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	6.16986	2.02250	1.82264	1.391
		10000	0	1.23786	0.20622	0.46901	14.465
	E	1000	0	3.70997	1.730831	0.89514	1.448
		10000	0	1.23648	0.37081	0.57610	14.507
10	R	1000	0	11.1945	4.27408	2.95424	2.768
		10000	0	3.7387	1.4517	0.9856	28.221
	E	1000	0	12.3623	4.17227	2.93762	2.833
		10000	0	02.4863	0.9922	0.9443	28.451
30	R	1000	10.0644	45.115	28.76103	7.47784	8.41883
		10000	4.23932	19.3346	10.2431	3.915104	84.324
	E	1000	9.28699	48.8435	25.06786	9.507925	8.324
		10000	4.00942	17.4161	10.11461	3.81334	84.331

Table 25: Rastrigin's function. Population 200. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0.000226762	2.4838	0.66190	0.77964	0.868
		10000	4e-5	1.23757	0.08405	0.31349	9.564
	E	1000	0	2.47201	0.7827	0.8878	0.879
		10000	0	2.47167	0.28837	0.62286	9.539
10	R	1000	4.24688	13.8916	8.2276	2.699	1.829
		10000	3.10801	12.2861	6.62671	2.43623	18.649
	E	1000	0.7369	11.3394	4.77417	3.04361	1.773
		10000	0.213861	3.48877	1.242717	0.79220	18.413
30	R	1000	149.012	213.559	178.094	16.6952	0
		10000	151.219	217.571	179.5222	017.21008	54.279
	E	1000	129.49	178.498	153.7466	12.8095	5.374
		10000	125.353	179.814	153.4707	11.4841	55.162

Table 26: Rastrigin's function. Population 100. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	2e-5	4.96153	0.45425	1.00239	1.873
		10000	2e-5	1.23739	0.04165	0.22583	18.991
	E	1000	0	1.2378	0.32962	0.55595	1.850
		10000	0	1.23584	0.04119	0.2256	18.931
10	R	1000	2.17334	8.84327	4.896658	1.62609	3.780
		10000	1.07972	6.53882	3.77150	1.27174	37.573
	E	1000	0.19927	8.70614	2.8665	2.2099	3.632
		10000	0.18295	0.96521	0.50493	0.19816	36.550
30	R	1000	123.213	189.285	156.50936	14.103721	10.717
		10000	124.624	0202.596	0163.8809	018.382	109.60
	E	1000	94.107	164.206	132.166	15.70035	11.013
		10000	118.05	157.246	136.3946	10.5377	108.771

Table 27: Rastrigin's function. Population 200. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	4.92341	1.3999	1.28542	1.063
		10000	0	2.47165	0.82388	0.87886	10.242
	E	1000	0	3.70747	1.27701	1.1466	1.057
		10000	0	3.70747	1.02985	1.17390	10.412
10	R	1000	0	12.3185	4.14596	2.90730	1.91
		10000	0	6.17926	2.47166	01.45142	19.492
	E	1000	0	11.0827	4.67151	3.01899	1.91856
		10000	0	6.17914	2.1420	1.51993	19.244
30	R	1000	38.1732	64.8228	51.4465	7.16074	5.536
		10000	33.8836	59.572	42.8753	6.4801	55.424
	E	1000	28.969	67.6087	44.42737	8.3098	5.477
		10000	20.0223	44.1675	29.44718	5.677646	55.845

Table 28: Rastrigin's function. Population 100. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	3.70747	0.90626	0.91411	2.082
		10000	0	2.47165	0.45313	0.75996	20.854
	E	1000	0	4.92341	0.864413	1.26100	2.079
		10000	0	2.47165	0.37074	0.66114	20.541
10	R	1000	0	6.17912	2.3892	1.652	3.7958
		10000	0	3.70747	1.23582	1.02629	39.070
	E	1000	0	8.63088	2.79987	1.91534	3.84093
		10000	0	4.94329	1.60656	1.26321	38.666
30	R	1000	28.1596	54.1222	38.27772	7.44445	11.021
		10000	24.0303	39.3919	30.4737	3.62003	111.634
	E	1000	23.1432	51.7928	35.4086	06.28163	11.105
		10000	11.5961	27.7319	18.84358	3.60846	110.393

Table 29: Rastrigin's function. Population 200. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	7.39569	2.14014	1.80004	0.880
		10000	0	4.94393	0.98868	1.14291	9.4290
	E	1000	0	6.15924	1.35807	1.66199	0.92656
		10000	0	3.70747	0.8238	1.0432	9.467
10	R	1000	1.24461	18.6696	6.63286	3.93677	1.757
		10000	0	11.2868	3.44543	2.37900	18.3411
	E	1000	0	17.6901	5.17721	4.3217	1.805
		10000	0	8.65344	2.76613	1.6466	18.216
30	R	1000	43.115	80.749	64.411	9.2469	5.410
		10000	24.6597	66.6703	43.81157	10.30359	54.8576
	E	1000	39.0904	71.8314	56.07896	7.693113	5.296
		10000	19.6891	50.0474	34.7075	7.28760	54.767

Table 30: Rastrigin's function. Population 100. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	2.47165	0.74149	0.69606	1.842
		10000	0	1.23582	0.20597	0.468436	19.068
	E	1000	0	2.47165	0.90626	0.91411	1.831
		10000	0	2.47165	0.49432	0.69606	18.918
10	R	1000	0	16.8438	3.26068	3.30798	3.499
		10000	0	4.96189	2.0656	1.275	36.729
	E	1000	0	9.88327	2.88446	2.20757	3.628
		10000	0	3.71153	1.31913	1.2550	37.044
30	R	1000	35.7065	74.8751	55.6209	9.33955	10.6573
		10000	22.7985	53.816	36.14355	8.34212	111.326
	E	1000	32.2477	72.6688	52.8739	10.5313	10.980
		10000	19.9112	47.4114	32.730	6.9044	111.028

Table 31: Rastrigin's function. Population 200. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	2.47173	0.741502	0.69607	0.942
		10000	0	2.47179	0.576732	0.842178	9.348
	E	1000	0	2.47165	0.94746	0.77370	0.921
		10000	0	2.47165	0.61791	0.77822	9.384
10	R	1000	1.2589	12.163	6.49084	2.9391	1.754
		10000	1e-5	7.60593	2.80613	02.07372	18.299
	E	1000	0	12.1473	4.6141	3.2511	1.8547
		10000	0	6.25973	2.61520	1.7152	18.420
30	R	1000	61.8625	117.239	89.45779	12.18508	5.276
		10000	75.4772	124.637	95.19794	11.2647	55.402
	E	1000	61.8294	107.378	80.68672	10.762879	5.432
		10000	61.3818	98.3481	79.79041	9.84086	55.613

Table 32: Rastrigin's function. Population 100. Elitist 10% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	0	2.47191	0.53553	0.7737	1.836
		10000	0	1.23594	0.24717	0.502793	18.602
	E	1000	0	1.23582	0.3295	0.55584	1.862
		10000	0	1.23582	0.28835	0.531628	18.814
10	R	1000	0.001768	9.27334	3.86537	2.0534	3.628
		10000	5e-5	6.2222	2.196520	1.929	37.147
	E	1000	3e-5	9.92494	2.02883	2.174935	3.611
		10000	0	4.98049	1.3667903	1.320420	37.156
30	R	1000	61.5287	104.474	80.7117	9.91440	11.084
		10000	52.1792	80.84757	12.75810	78.593	78.1657
	E	1000	51.9528	94.6591	71.2116	10.068	10.947
		10000	49.0876	78.6627	64.52396	7.61233	110.703

Table 33: Rastrigin's function. Population 200. Elitist 10% Selection.

3.4 Michalewicz's function

The Michalewicz function is a multimodal test function (n! local optima). The parameter m defines the "steepness" of the valleys or edges. Larger m leads to more difficult search. For very large m the function behaves like a needle in the haystack (the function values for points in the space outside the narrow peaks give very little information on the location of the global optimum).

$$f(x) = -\sum_{i=1}^{n} sin\left(x_i\right) \cdot \left(sin\left(\frac{i \cdot {x_i}^2}{\pi}\right)\right)^{2\cdot m} \qquad x_i \in [0, \pi]\,,\ i = \overline{1, n},\ m = 10$$

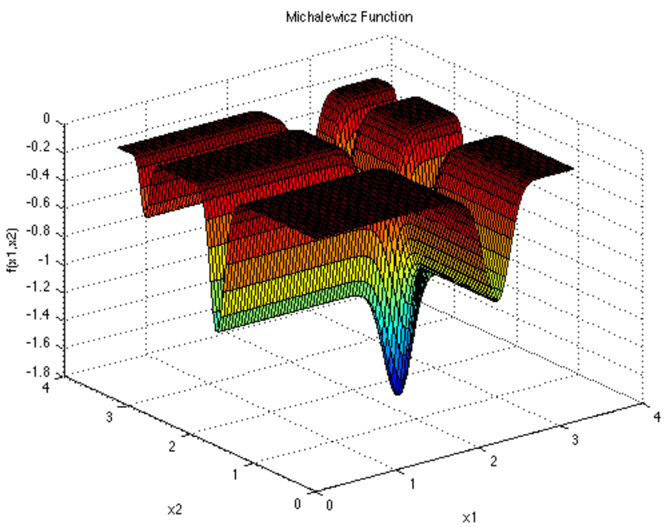

Figure 4: Michalewiczs's function[10]

3.4.1 Fitness function

Let f_{12} be Michalewicz's function and M_{12} its fitness function.

Dimensions	Global minimum
5	-4.68765
10	-9.66015
30	-29.63088

Table 34: Michalewicz's function known global minima[6][13]

$$0.734375^{10 \cdot f_{12}(x)} \; ; \; x \in [0, \pi]^n \; ; \; n \geq 1$$

It is known that Michalewicz's function codomain is $\mathbb{R}_-$. Because of this, the fitness function that was chosen to be an exponential function with a subunit base. The base was found empirically. Through tests it was observed that the selection pressure wasn't big enough, so the exponent of the fitness function was multiplied by a constant which was found empirically.

3.4.2 Results

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68665	-4.04753	-4.52338	0.16452	0.682
		10000	-4.68697	-4.37131	-4.56984	0.100284	6.864
	E	1000	-4.68716	-4.28968	-4.539941	0.1184	0.695
		10000	-4.68732	-4.33184	-4.618409	0.09218	6.860
10	R	1000	-9.54396	-8.45847	-9.003703	0.256	1.3220
		10000	-9.60795	-9.05704	-9.34292	0.1508	13.226
	E	1000	-9.53408	-8.58867	-9.14373	00.2455	1.3223
		10000	-9.64323	-8.8229	-9.3521	0.18416	12.970
30	R	1000	-27.4044	-25.3344	-26.4533	0.5097	3.954
		10000	-28.3188	-26.6559	-27.661	0.4234	38.595
	E	1000	-27.7212	-25.7594	-26.55502	0.45451	03.897
		10000	-28.2843	-26.1553	-27.61295	0.49726604	38.261

Table 35: Michalewicz's function. Population 100. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68646	-4.20071	-4.58536	00.11489	1.326
		10000	-4.68606	-4.37135	-4.608637	0.08860	13.050
	E	1000	-4.68756	-4.28938	-4.54350	0.1263	1.352
		10000	-4.68746	-4.3746	-4.6063	0.0989	13.461
10	R	1000	-9.46672	-8.69141	0-9.17246	0.2047	2.578
		10000	-9.65301	0-9.01823	-9.45302	0.14242	025.749
	E	1000	9.56896	-8.6616	-9.162	0.21955	2.592
		10000	-9.65496	-9.12065	-9.447157	0.1420	26.0911
30	R	1000	-27.905	-26.3003	-27.05574	0.4134	7.587
		10000	-28.9833	-27.2146	-27.941	0.4218	76.550
	E	1000	-27.9482	-25.6845	-27.03641	0.58822	7.730
		10000	-28.6328	0-27.3921	-28.15292	0.29416	76.130

Table 36: Michalewicz's function. Population 200. Roulette Wheel Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68765	-4.21297	-4.557588	0.12792	0.838
		10000	-4.68766	-4.53322	-4.647280	0.06315	9.186
	E	1000	-4.68766	-4.34584	-4.554782	0.106554	0.888
		10000	-4.68766	-4.52099	-4.6437736	0.067308	9.217
10	R	1000	-9.39885	0-8.73731	-9.08253	0.2013	1.7639
		10000	-9.47251	-8.87158	-9.295978	0.13707	17.460
	E	1000	-9.5514	-8.59084	-9.236962	0.221073	1.728
		10000	-9.60014	-8.89786	-9.4298	0.1431	17.930
30	R	1000	-18.0847	-15.4588	-16.7429	0.688116	5.209
		10000	-19.6651	-14.9498	-16.9432	0.94686	52.800
	E	1000	-20.5018	-16.3653	-18.22425	0.96598	5.155
		10000	-19.972	-17.0553	-18.51653	0.7554	52.713

Table 37: Michalewicz's function. Population 100. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68766	-4.49589	-4.65043	0.06114	1.750
		10000	-4.68766	-4.53765	-4.656352	0.06040	18.609
	E	1000	-4.68766	-4.44861	-4.622058	0.072588	1.820
		10000	-4.68766	-4.53325	-4.66610466	0.051781995	18.507
10	R	1000	-9.4939	-8.45033	-9.265648	0.196594	3.490
		10000	-9.60844	-9.23095	-9.42838	0.09293	35.539
	E	1000	-9.56956	-9.15147	-9.37648	0.11064	3.506
		10000	-9.61866	-9.35563	-9.521280	0.0752699	35.648
30	R	1000	-19.6246	-16.5844	-18.1107	0.819287	10.358
		10000	-18.787	-16.8849	17.96740	0.499602	105.336
	E	1000	-21.9677	-18.3116	-19.73194	0.836456	10.483
		10000	-21.4766	-18.0099	-19.5414	0.667850	105.464

Table 38: Michalewicz's function. Population 200. Tournament Size 2 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68759	-4.00497	-4.54115	0.14663	1.004
		10000	-4.68766	-4.3749	-4.619496	0.08532	10.108
	E	1000	-4.68259	-4.22532	-4.519671	0.10882	1.009
		10000	-4.68766	-4.49147	-4.627467	0.07495	10.125
10	R	1000	-9.64816	-8.73776	-9.17288	0.216054	1.828
		10000	-9.57622	-9.06254	-9.38931	0.12583	18.712
	E	1000	-9.5154	-8.64455	-9.1223	0.2199	1.827
		10000	-9.6174	-9.31319	-9.479513	0.08147	18.750
30	R	1000	-26.6512	-23.6535	-25.11856	0.792054	5.348
		10000	-26.3926	-24.4237	-25.4352	0.52070	53.557
	E	1000	-27.2871	-23.229	-25.4666	0.8267	5.278
		10000	-26.9416	-24.2861	-25.8696	0.6014	53.591

Table 39: Michalewicz's function. Population 100. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68766	-4.46386	-4.625177	0.073633	1.953
		10000	-4.68766	-4.51659	-4.6387	0.07057	20.255
	E	1000	-4.68766	-4.35636	-4.5853	0.102367	1.996
		10000	-4.68766	-4.53325	-4.65361	0.0599974	20.379
10	R	1000	-9.62079	-8.74605	-9.331547	0.2154	3.662
		10000	-9.66015	-9.3137	-9.5138	0.088501	37.583
	E	1000	-9.6152	-8.78137	-9.30714	0.22576	3.666
		10000	-9.6584	-9.21681	-9.475173	0.1115605	37.763
30	R	1000	-27.2777	-24.7643	-26.1514	0.6079	10.656
		10000	-27.0252	-25.5022	-26.35791	0.46883292	107.670
	E	1000	-27.4986	-25.3341	-26.377	0.5779	10.677
		10000	-27.7044	-26.2039	-26.99839	0.423504	107.657

Table 40: Michalewicz's function. Population 200. Tournament Size 7 Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68766	-4.3452	-4.57914	0.1103	0.8911
		10000	-4.68766	-4.37489	-4.63899	0.0843	9.076
	E	1000	-4.68765	-4.28955	-4.57317	0.10611	0.89566
		10000	-4.68766	-4.3749	-4.62988	0.0842	9.078
10	R	1000	-9.54987	-8.59083	-9.145465	0.21174	1.751
		10000	-9.61339	-8.72356	-9.285025	0.193184	17.501
	E	1000	-9.52844	-8.56709	-9.128245	0.2367390	1.756
		10000	-9.56823	-8.98266	-9.32155	0.171135	17.531
30	R	1000	-26.0674	0-22.8548	-24.3669	0.727903	5.149
		10000	-25.8179	-23.1063	-24.66749	0.654818	52.512
	E	1000	-25.979	-23.0792	-24.7661	0.6905	5.136
		10000	-25.9973	-24.2793	-25.1664	0.487865	52.5813

Table 41: Michalewicz's function. Population 100. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68766	-4.3749	-4.64517	0.07831	1.750
		10000	-4.68766	-4.53766	-4.6623133	00.05670	17.963
	E	1000	-4.68766	-4.3749	-4.623556	0.0931	1.833
		10000	-4.68766	-4.53766	-4.6576	0.0609	18.052
10	R	1000	-9.49366	-8.86034	-9.211764	0.165560	3.563
		10000	-9.64839	-8.90165	-9.36139	0.16933	35.611
	E	1000	-9.64814	-8.9994	-9.40096	0.1661504	3.482
		10000	-9.65964	-9.04659	-9.4806	0.13655	35.201
30	R	1000	-26.5284	-23.9703	-25.2395	0.5925	10.443
		10000	-26.5367	-24.7186	-25.5321	0.49315	105.576
	E	1000	-26.3994	-24.3434	-25.442	0.503330	10.502
		10000	-27.2008	-25.0651	-26.17956	0.53231	104.842

Table 42: Michalewicz's function. Population 200. Elitist 4% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68593	-4.35822	-4.572893	0.094346	0.881
		10000	-4.68766	-4.5249	-4.65609	0.0613781	8.979
	E	1000	-4.68766	-4.3742	-4.586552	0.088089	0.8819
		10000	-4.68766	-4.5249	-4.65376	0.0614662	9.034
10	R	1000	-9.41675	-8.55575	-9.11505	0.240820	1.711
		10000	-9.5997	-8.97341	-9.3583	0.165	17.440
	E	1000	-9.55137	-8.77189	-9.19432	0.17644	1.743
		10000	9.61343	-8.94968	-9.337307	0.163135	17.7265
30	R	1000	-23.3127	-20.4537	-22.098	0.70315	5.119
		10000	-23.6735	-21.2548	-22.3896	0.67167	52.482
	E	1000	-24.215	-21.276	-23.04	0.77027	5.155
		10000	-25.5612	-21.9482	-23.24179	0.8585737	52.5155

Table 43: Michalewicz's function. Population 100. Elitist 10% Selection.

D	CX	G	Min	Max	μ	σ	μ_δ
5	R	1000	-4.68766	-4.5249	-4.66198	0.052928	1.808
		10000	-4.68766	-4.53765	-4.671961	0.045549	18.079
	E	1000	-4.68766	-4.5249	-4.65569	0.0564577	1.770
		10000	-4.68766	-4.53766	-4.6667316	0.0515055	18.182
10	R	1000	-9.55143	-8.64583	-9.3231	0.20839	3.4144
		10000	-9.60025	-9.2031	-9.43394	0.110172	35.136
	E	1000	-9.63348	-8.88812	-9.39371	0.18163	3.529
		10000	-9.62005	-8.9927	-9.4740	0.1391	35.263
30	R	1000	-24.4506	-21.8502	-22.944213	0.652307	10.276
		10000	-24.2357	-22.429	-23.264	0.54137	105.508
	E	1000	-24.9792	-22.5553	-23.8291	0.62984	10.469
		10000	-24.6402	-22.6449	-23.85566	0.596632	105.788

Table 44: Michalewicz's function. Population 200. Elitist 10% Selection.

4 Observations

A first observation to be made is for the difference of the crossover methods. Throughout the tables, the *elitist* crossover method has gotten better results while maintaining a close execution time to the *replacement* crossover method. The better results come from the fact that after the crossover operation, the population is guaranteed to be at least as fit as it was before the operation. This scenario happens when the offspring have a fitness level lower than their parents. This guarantees that fit individuals are not lost after the operation whereas in the *replacement* method, the offspring always replace the parents which can lead to the loss of well fit individuals.

The different sizes of the population affect the processing time and they also affect the quality of the final result. A higher count of individuals will give the genetic algorithm more chances to escape from a local optimum. This can be seen in the tables 24 and 25. A higher number of individuals results in a higher count of crossover operations between the individuals resulting in the chance of generating better fit individuals. This can be seen in Schwefel's function tables 15 and 16.

For the *tournament* selection method, the size of a tournament directly affects the selection pressure as in a tournament with many participants, less fit individuals would contest the position in the population with more fit individuals. This behaviour can be seen between the tables with a tournament size of 2 and a tournament size of 7. Size 2 is the minimum size of a tournament and since only tournament winner are selected into the new population, there is a higher chance of less fit individuals to get selected. This is less likely in the opposite case where the size of the tournament is higher. This can be observed easily in Schwefel's function tables 15 and 17.

For the *elitist* selection method, it is built on top of the *Roulette Wheel* selection method. For this experiment, it has been decided that a percentage p of the population would be kept for the crossover operation. The percentage p belongs into $\{4, 10\}$ and is used to point that if it is opted in favor of adding elitism to the selection scheme, a low percentage of individuals have to be kept for the new population because a high percentage would guarantee the safety of those chromosomes. Even though they might be in the top $p\%$ of the current population they might not be fit enough to support the genetic algorithm towards its goal.

5 Comparison to trajectory strategies

In a previous report[27], I have compared two common trajectory algorithms: *Hill Climbing* and *Simulated Annealing*. The latter had a great waiting time compared to the former and it also had better results while being more flexible thanks to the temperature component. Because of it, I concluded that the *Simulated Annealing* algorithm can be tweaked for even better results. I'll use Schwefel's function to compare the results of the genetic algorithms used in this paper to the results of the *Simulated Annealing* algorithm. For the comparison, the population count will be 200, the generation count 10000 and the crossover method the *elitist* one. For the *tournament* selection method, the size of the tournament will be **7** and for the *elitist* selection method the percentage will be **4%**.

Dimensions	Method	Expected Result	μ_{value}	σ_{value}	μ_{time}
5	WoF	-2094.9145	-2094.6493	0.15065	19.249
	T_7		-2094.62366	0.10736	10.504
	E_4		-2094.75533	0.10546	25.257
	SA		-2094.74166	0.09512	29.187
10	WoF	-4189.829	-4188.75866	0.31439	38.070
	T_7		-4189.267	0.17620	19.940
	E_4		-4189.25833	0.20525	50.131
	SA		-4176.41466	26.67730	43.371
30	WoF	-12569.487	-12551.22	5.30714	113.7873
	T_7		-12489.48	21.77849	57.556
	E_4		-12105.84666	207.18577	151.620
	SA		-12550.72666	31.07277	81.102

Table 45: Comparison for Schwefel's function

We can observe that the genetic algorithm has better all around results compared to the *Simulated Annealing* algorithm. The genetic has a lower waiting time while being given a high count of individuals and generations.

6 Conclusions

Genetic algorithms have more freedom in regard to finding the optimum of a function compared to trajectory methods. Its parameters can be easily modified to get the desired result. The approach towards the crossover and selection methods has an impact on the quality of the solution.

References

[1] Eugen Croitoru, Teaching: Genetic Algorithms
https://profs.info.uaic.ro/~eugennc/teaching/ga/

[2] Eugen Croitoru, LaTeXexample
https://gitlab.com/eugennc/teaching/-/tree/master/GA

[3] De Jong's function
http://www.geatbx.com/docu/fcnindex-01.html#P89_3085

[4] Schwefel's function
http://www.geatbx.com/docu/fcnindex-01.html#P150_6749

[5] Rastrigin's function
http://www.geatbx.com/docu/fcnindex-01.html#P140_6155

[6] Michalewicz's function
http://www.geatbx.com/docu/fcnindex-01.html#P204_10395

[7] De Jong's function graph
https://www.sfu.ca/~ssurjano/spheref.html

[8] Schwefel's function graph
https://www.researchgate.net/figure/Two-dimension-Schwefel-function_fig1_311993569

[9] Rastrigin's function graph
https://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/rastrigin.html

[10] Michalewicz's function graph
https://www.sfu.ca/~ssurjano/michal.html

[11] Value of π
https://en.wikipedia.org/wiki/Pi

[12] Text merger
https://www.filesmerge.com/merge-text-files

[13] Charlie Vanaret, Jean-Baptiste Gotteland, Nicolas Durand, Jean-Marc Alliot, *Certified Global Minima for a Benchmark of Difficult Optimization Problems*, September 2014
https://www.researchgate.net/publication/278769271_Certified_Global_Minima_for_a_Benchmark_of_Difficult_Optimization_Problems

[14] Uniform int distribution *C++*
http://www.cplusplus.com/reference/random/uniform_int_distribution/

[15] Uniform real distribution *C++*
http://cplusplus.com/reference/random/uniform_real_distribution/

[16] Multithreading in *C++*
https://www.geeksforgeeks.org/multithreading-in-cpp/

[17] *C++* Mersenne twister random number generator
https://www.cplusplus.com/reference/random/mt19937_64/

[18] Mersenne twister example
https://www.guyrutenberg.com/2014/05/03/c-mt19937-example/

[19] Prof. Dr. Henri Luchian, Prep. Drd. Mihaela Breabăn, *Algoritmi Genetici*, Univestitatea "Alexandru Ioan Cuza", Facultatea de Informatică, Departamentul de Învățământ la Distanță, 2006-2007
http://students.info.uaic.ro/~vladut.ungureanu/Algoritmi-genetici-ID.pdf

[20] Roulette Wheel Selection
https://en.wikipedia.org/wiki/Fitness_proportionate_selection

[21] Calvin McGowan, *Genetic Algorithms: Tournament Selection*, 9^{th} March 2017
https://bit.ly/36lLwnG

[22] Selection - Writing a Genetic Algorithm from scratch
https://www.youtube.com/watch?v=pJ45IaGzI1c

[23] S Prayudani, A Hizriadi, E B Nababan, S Suwilo, *Analysis Effect of Tournament Selection on Genetic Algorithm Performance in Traveling Salesman Problem (TSP)*, Journal of Physics: Conference Series, IOP Publishing Ltd
https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012131/pdf

[24] Tournament selection
https://en.wikipedia.org/wiki/Tournament_selection

[25] http://www.lendek.net/teaching/opt_en/ga.pdf

[26] Huayang Xie, Mengjie Zhang, *Sampling Issues of Tournament Selection in Genetic Programming*
https://ecs.wgtn.ac.nz/foswiki/pub/Main/TechnicalReportSeries/ECSTR09-07.pdf

[27] Tudose George-Daniel, *Comparison of Hill Climbing variants and Simulated Annealing algorithms on a set of functions*, November 2020
https://github.com/BeterNerfIrelia/GeneticAlgorithms/blob/main/Homework%20T1/GAHomework1/Tudose_George-Daniel_Homework1_GA.pdf

[28] Force text to appear after table
https://tex.stackexchange.com/questions/150634/how-to-force-a-text-to-appear-after-a-table