Machine Learning - Assignment 5

Brian Pulfer

December 2019

1 Visualizing the test functions

1.1 a

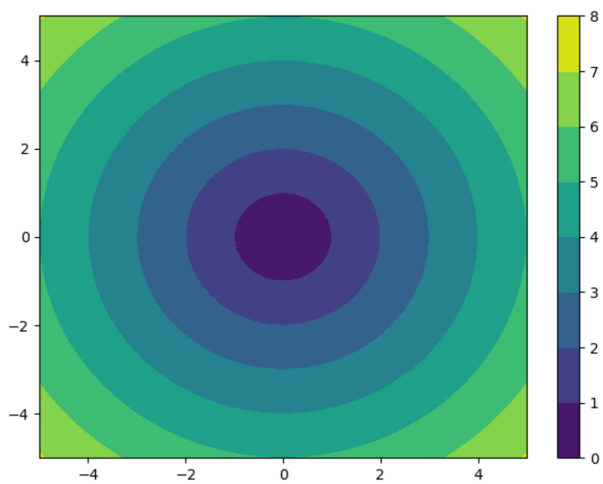

Figure 1: Sphere function's contour plot for n = 2

By figure 1, we can see that the sphere function for 2 dimensions in the given domain ([-5, 5]) has this attitude: in the origin there's a minimum, and as we get far away from the origin, the function value increases.

1.2 b

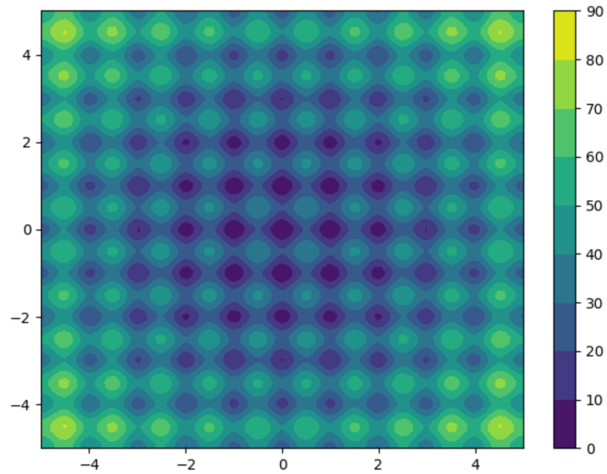

Figure 2: Rastrigin function's contour plot for n = 2

In figure 2 we instead see the rastrigin function for 2 dimensions in the given domain ([-5, 5]). We can see that in this function, opposed to the sphere function, we have a multitude of local minima. The global minima however, is still in the origin.

1.3 c

Points for both functions where sampled linearly from one extreme of the domain to the other (from -5 to 5 for both coordinates). Every combination of coordinates was tested, thus 100x100 = 10'000 points were sampled uniformly. After evaluating all of them, for both functions the minimum seems to be in the origin (point (-0.05, -0.05) for both functions).

2 CEM

2.1 a

CEM was run 3 times on both test functions at high dimensionality (100 dimensions). The initial population was randomly uniformly initialized in the given domain ([-5, 5]) for every coordinate.

2.2 b & c

After tuning the parameters a good performance for both functions can be found with: 200 generations with 500 individuals and an elite rate of the 30%. With these parameters, the best final fitness for the sphere function is around 0.53 (note that we are minimizing the functions) and the best final fitness for the rastrigin function is around 100.

We can note how the rastrigin function is harder to optimize, due to the fact that many local optima are presents and, unlike the sphere function, the search could get stuck in one of these minima.

2.3 d

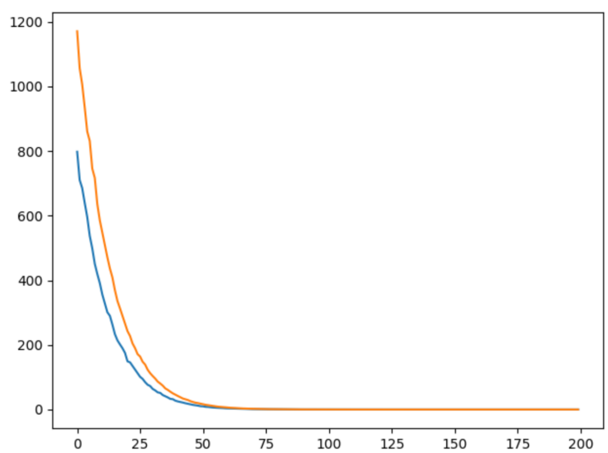

Figure 3: CEM algorithm best and worst evaluations for sphere function

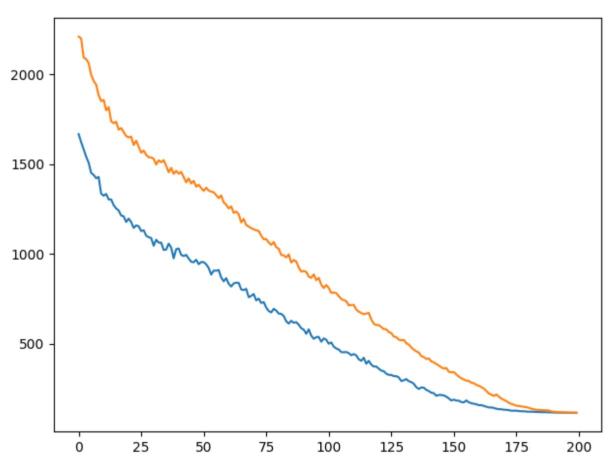

Figure 4: CEM algorithm best and worst evaluations for rastrigin function

In figure 3 and 4 we see the best and worst evaluations for each of the 200 generations (x-axis). Note that the best evaluations are the lowest ones, since we're minimizing the function value.

We can see in both plots that the algorithm seems to converge when best and worst individuals evaluations meet. Also, the algorithm does a good job at minimizing the function generation by generation.

3 NES

3.1 a

NES was run 3 times on both test functions at high dimensionality (100 dimensions). The initial population was randomly uniformly initialized with values within 5-10 for every coordinate.

3.2 b & c

After tuning the parameters, one can soon find out that this algorithm requires much more generations to get close to the global minima.

The plots that are going to be showed in the next point were obtained with 2'000 generations with 100 individuals each and a learning rate of 0.00001.

3.3 d

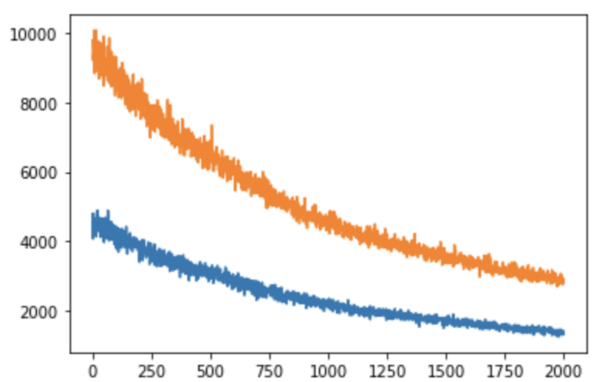

Figure 5: NES algorithm best and worst evaluations for sphere function

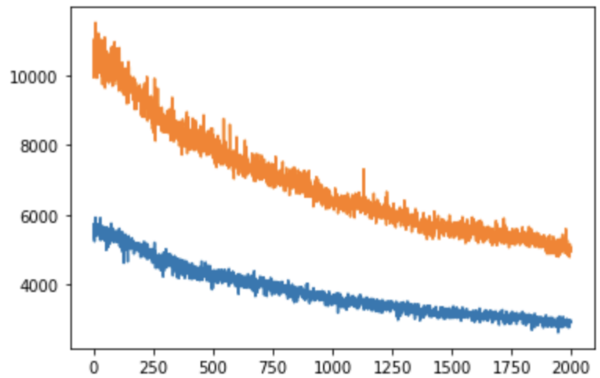

Figure 6: NES algorithm best and worst evaluations for rastrigin function

In figure 5 and 6 we see that the behaviour of the algorithm is to create more and more fit individuals every as generations advance. By letting the algorithm run even further, we'll surely find that best and worst candidates have very similar fitness in very high generations.

The final best fitness is around 1'330 for the sphere function, while it's around 2'900 for the rastrigin function.

4 CMA-ES

4.1 a

CMA-ES was run 3 times on both test functions at high dimensionality (100 dimensions). The initial population was randomly uniformly initialized in the given domain ([-5, 5]) for every coordinate.

4.2 b & c

After tuning the parameters a good performance for both functions can be found with: 250 generations with 1'000 individuals each and an elite rate of the 40%. With these parameters, the best final fitness for the sphere function is around 1e-4 and the best final fitness for the rastrigin function is around 300.

4.3 d

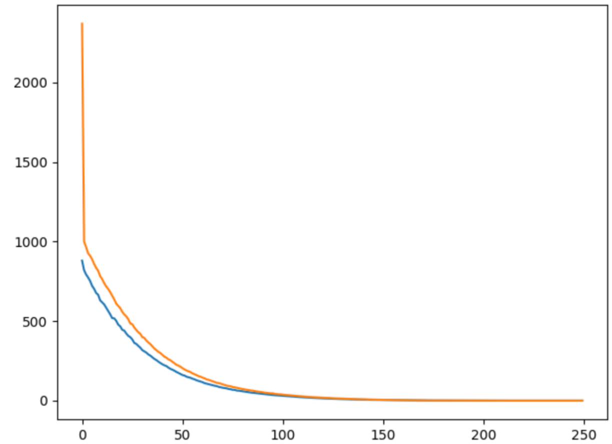

Figure 7: CMA-ES algorithm best and worst evaluations for sphere function

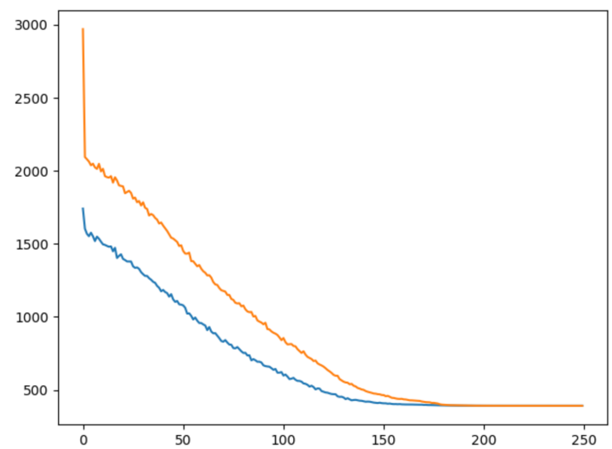

Figure 8: CMA-ES algorithm best and worst evaluations for rastrigin function

Once more, we see in figure 7 and 8 the behaviour of the best and worst candidates over generations using the CMA-ES algorithm with the aforementioned parameters.

We see that the behaviour over generation is still the same: best and worst individuals get better and better fitness's as generations go on until they eventually converge to the same value (or very very close).

5 Benchmarking

5.1 a

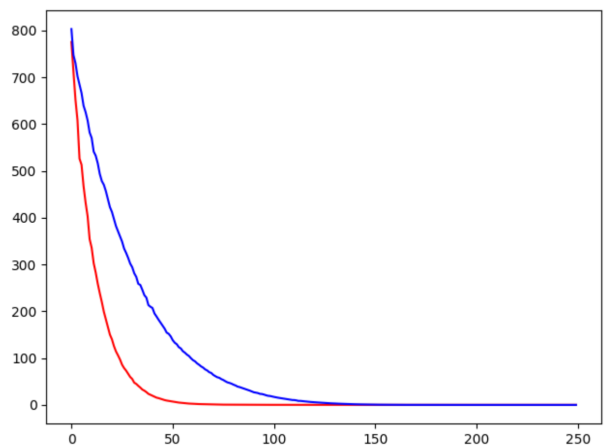

Figure 9: CEM and CMA-ES best individuals per generation (sphere)

In figure 9 we can see how the best individual perform in each of the 250 generations with the algorithms CEM (red) and CMA-ES (blue). Note that the NES algorithm was excluded because it was not comparable to the other two (performed very poorly in so few generations)

By the figure, we can see how CEM has a better fitness in every generation (considering the best individual).

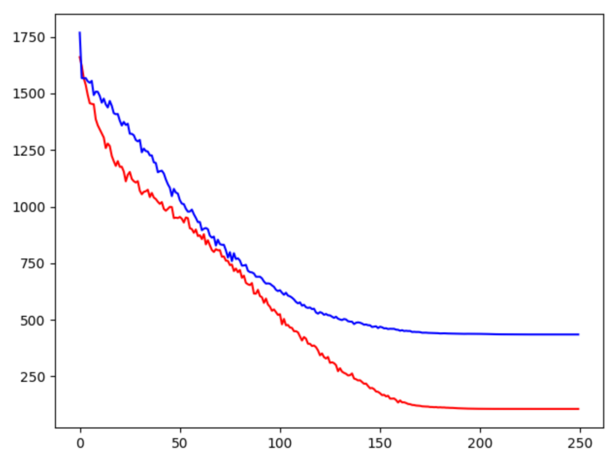

Figure 10: CEM and CMA-ES best individuals per generation (rastrigin)

In figure 10 we see the same plot applied to the rastrigin function. Once again, the CEM algorithm seems to be performing better than the CMA-ES algorithm.

5.2 b

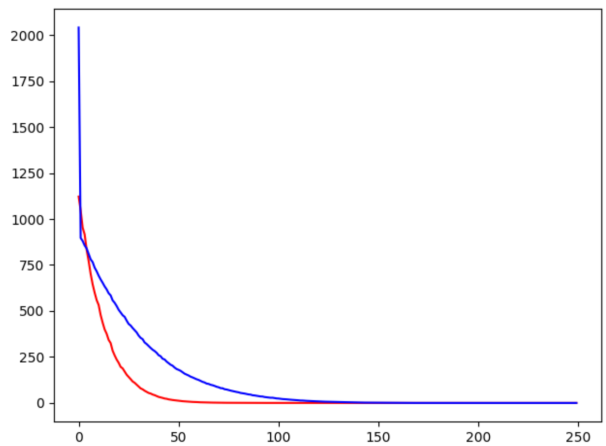

Figure 11: CEM and CMA-ES worst individuals per generation (sphere)

In figure 11 we can see how the worst individual perform in each of the 250 generations with the algorithms CEM (red) and CMA-ES (blue). Note that the NES algorithm was excluded because it was not comparable to the other two (performed very poorly in so few generations)

We see that the CEM algorithm seems to provide better candidates even when it comes to considering the worst individuals.

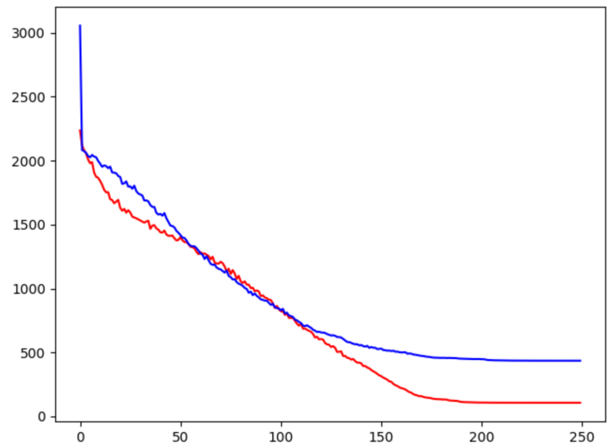

Figure 12: CEM and CMA-ES worst individuals per generation (rastrigin)

In figure 12 we see the same plot applied to the rastrigin function. Once more, we can clearly proclaim the CEM algorithm as winner.

5.3 c

For both functions (sphere and rastrigin), the CEM algorithm seems to perform slightly better than the CMA-ES. This depends however very much on the parameters of each algorithm and can vary by adjusting those parameters.