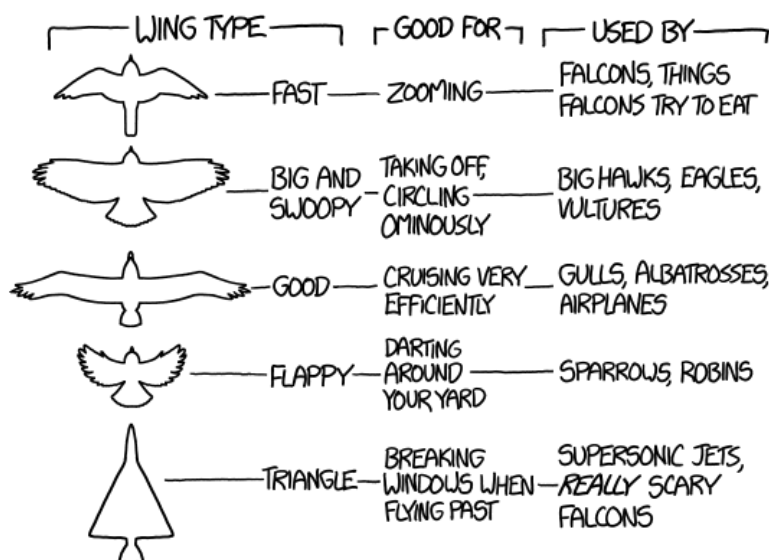


ECTA Homework 3

Shape Matching Problem

Debaraj Barua (9030412), Md Zahiduzzaman (9030432)

June 14, 2018



To optimize wings, it is useful to start with known designs and then adjust them to specific tasks. How to convert from one representation to another? In this assignment you will convert the 4-dimensional NACA wing specific representation into a general B-Spline representation with 32 control points. You will compare 2 evolutionary methods, Genetic Algorithms and Evolution Strategies.

1 Assignment Description

Shape Matching Problem

1. Write your own Genetic Algorithm to solve the shape matching problem
2. Write your own version of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to solve the shape problem.
3. Compare the performance of the two algorithms on three airfoils (NACA airfoil shapes: 0012, 5522, 9735). Is there a significant difference between a GA and an ES?

- Grading Scheme

- ☐ Genetic Algorithm (20 pts)
 - ☐ Bitstring *or* Real-Valued (20 pts)
 - ☐ Bitstring *and* Real-Valued (+10pts)
- ☐ Evolution Strategies (60 pts)
 - ☐ ES with 1/5 rule (30 pts)
 - ☐ CMA-ES with out evolution paths (30 pts)
 - ☐ CMA-ES with evolution paths (+10)
- ☐ Comparisons (20 pts)
 - ☐ Big beautiful wall of data

2 Submission Instructions

To be perfectly clear we expect two submissions to LEA:

1. 1 PDF (report) – a modified version of your submission PDF, with your own code snippets, figures, and responses inserted
2. 1 ZIP (code and data) – a .zip file containing all code use to run experiments (.m files) *and* resulting data as a .mat file
3. Make sure to follow the naming scheme
HW_NUMBER_LASTNAME1_LASTNAME2.suffix
4. → A valid name would be HW_02_Smith_Fernandez.pdf
5. Make sure both team members use the same filename!

3 The Assignment

3.1 Genetic Algorithm

- Genetic Algorithms are typically represented by a string, this string could take many forms, such as bitstrings or real-valued numbers. What are the advantages and disadvantages of each encoding in this application? How would you convert each into 32 real-valued numbers? How could you perform crossover and mutation in each? Which do you think would be best?

1. Bitstring

One way to represent the problem will be to represent each estimate for the Y value as the binary representation of the number.

The advantage of such an approach will be ease of mutation of the bit string by bit flips. On creating new individuals in this manner, we would however have to ensure each individual lies within the given range. Another advantage would be that we can avoid round off errors resulting from floating point operations using real valued numbers.

Crossover can be implemented as a single point crossover, where we create new individual from the two parts of each parent.

Mutation can be implemented by a bitflip according to a mutation probability.

2. Real-valued

Representation: A string of 32 real valued numbers between -0.5 to 0.5 can be used to represent the problem. These 32 numbers together are a representation of a single genome with possible values of the estimates of Y-value against the control points.

Advantage: These values can be used to represent the Y position of the 32 control points directly, this makes it easy to check the fitness by comparing with the actual value.

Disadvantage: Round off error in floating point operations.

Crossover: Crossover is achieved by taking two parent ids and then getting the mean of the their genomes depending on the crossover probability. This mean value is used to represent a new child genome.

Mutation: It is achieved by generating gene values between -0.5 and 0.5 to replace genes based on the mutation probability.

- Implement the encoding you think will perform best and plots its median performance over 20 iterations, using 20,000 evaluations.

Your plot here

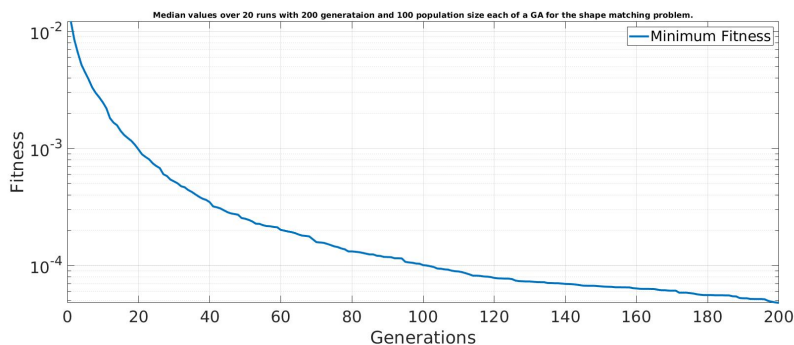


Figure 1: Median fitness values over 20 runs with 200 generations and 100 population size each of a GA for the shape matching problem.

- ****Extra Credit**** Implement both encodings and compare them on matching task for one of the shapes. Is one significantly better? Can you explain why?
 Your text here + a figure w/significance here

3.2 Evolution Strategies

- CMA-ES is an advanced version of ES. As a first step implement a simple ES first.
 - In this ES mutation of all parameters should have equal strength which is adjusted by the 1/5th rule.
(every N generations change mutation strength: if $> 1/5$ th of mutations resulted in an better fitness (i.e. a best solution) increase mutation strength, otherwise decrease mutation strength). Test your implementation on one of the shapes.

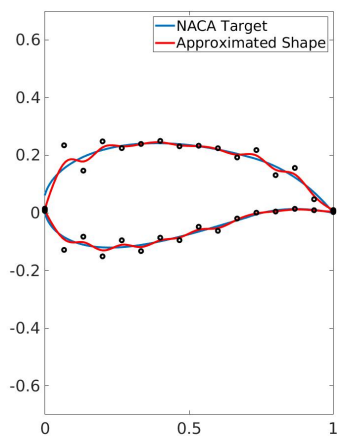


Figure 2: Plot on NACA [9, 7, 3, 5] with ES

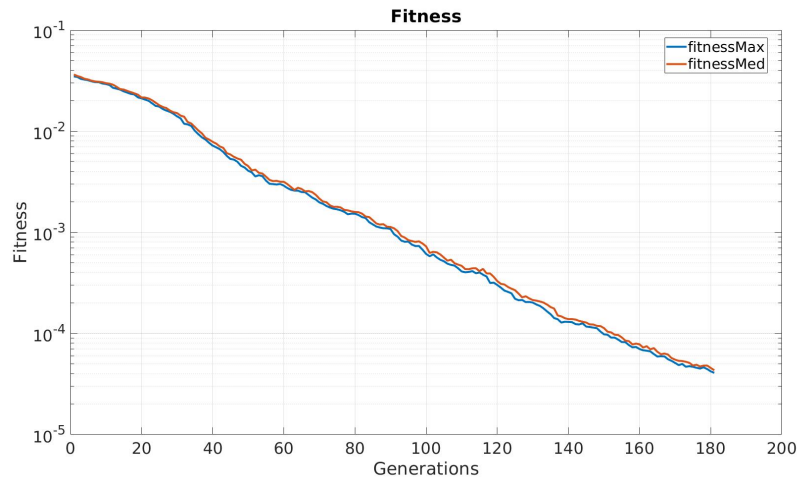
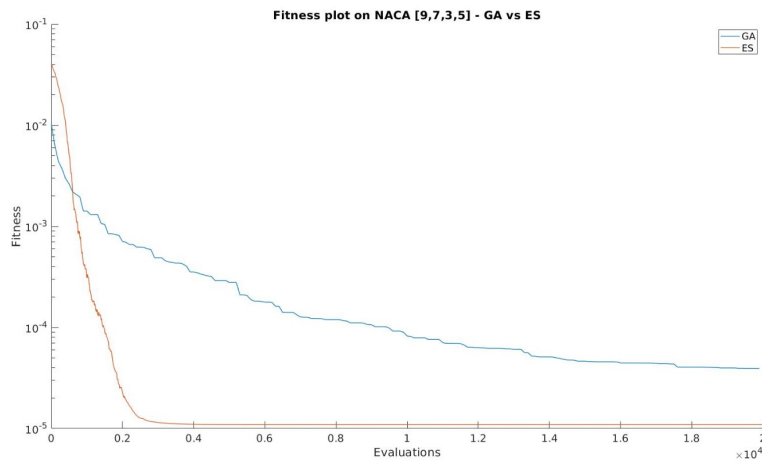


Figure 3: Fitness plot on NACA [9, 7, 3, 5] with ES

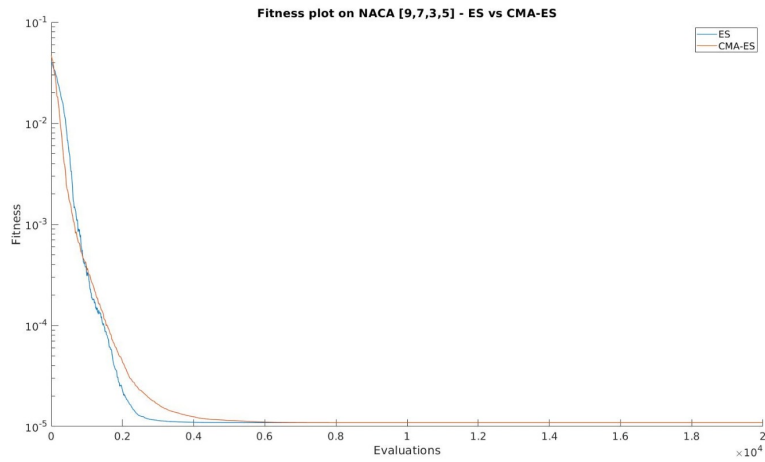
- Compare to your best GA. Is one significantly better than the other?



It is observed that the GA does not manage to converge to the global minima even after 20000 evaluations.

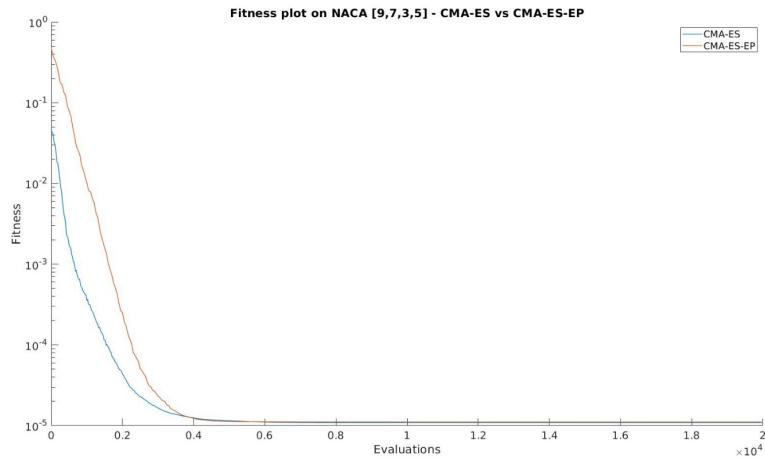
However, the ES manages to converge to the global minima within 2000-2500 evaluations. Thus it can be seen to be significantly better than the GA.

- Now program CMA-ES without evolution paths.
 - Compare to your ES results. Is there a significant improvement?



It is observed that both the ES and CMAES manages to converge to the global minima, however, the ES is seen to perform slightly better. This may be attributed to the selection of parameters, including the population size and number of generations.

- *****Extra Credit***** Now program CMA-ES with evolution paths.
 - Compare to your previous CMA-ES results. Is there a significant improvement?



CMAES-EP is seen to converge slightly faster than the CMAES. This can be attributed to the fact that the path of evolution adapts the way the step sizes are updated, thus leading to faster convergence.

3.3 Comparisons

Produce one plot which shows the performance of each algorithm. Run each algorithm 20 times on each shape (NACA airfoil shapes: 0012, 5522, 9735), with a budget of 20,000 function evaluations. For each run record the best ever found individual at each evaluation. For each algorithm plot the median fitness of this best ever individual. This may take some time, write a script to run and collect this data. The following should be on this plot (leaving out any algorithms you chose not to implement):

1. Binary GA
2. Real-Valued GA
3. ES
4. CMA-ES (without evolution paths)
5. CMA-ES (with evolution paths)

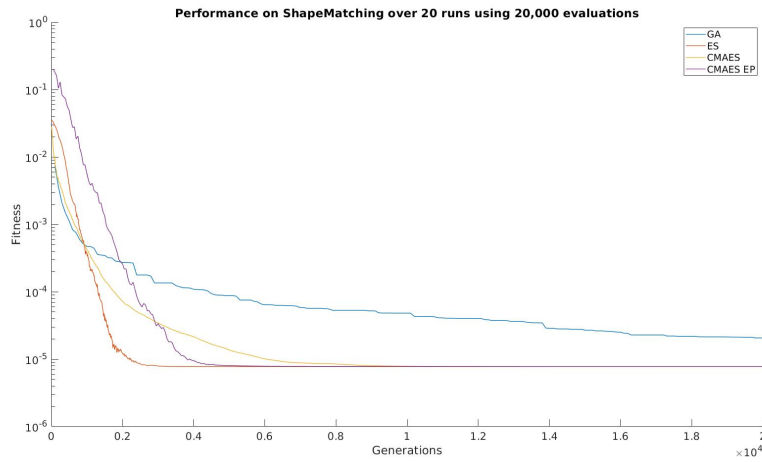


Figure 4: Best values across all air foils over 20 runs with 20,000 evaluations each for the GA, ES, CMAES and CMAES EP for the shape matching problem.