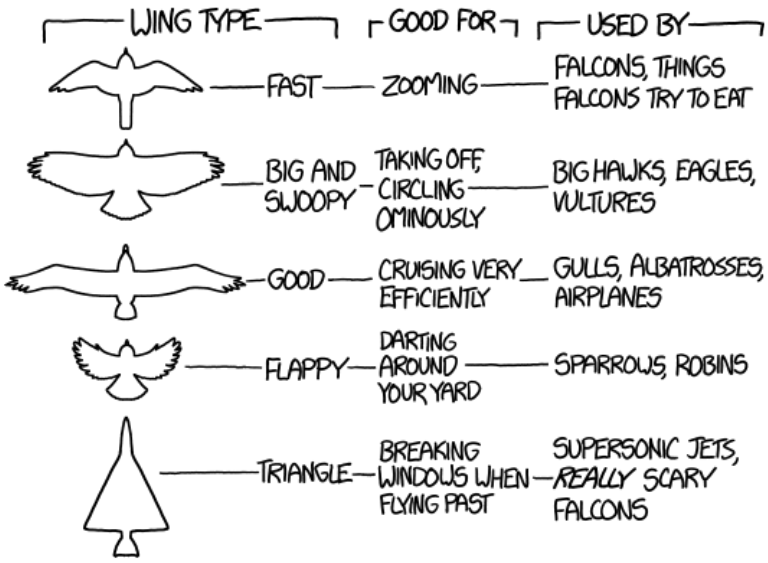
ECTA Homework 3
Shape Matching Problem

Djordje Vukcevic, djordje.vukcevic@smail.inf.h-brs.de
Supriya Vadiraj,supriya.vadiraj @smail.inf.h-brs.de

June 15, 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

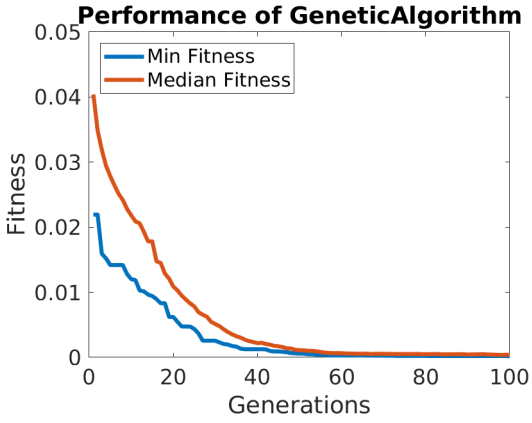
3.1 Genetic Algorithm

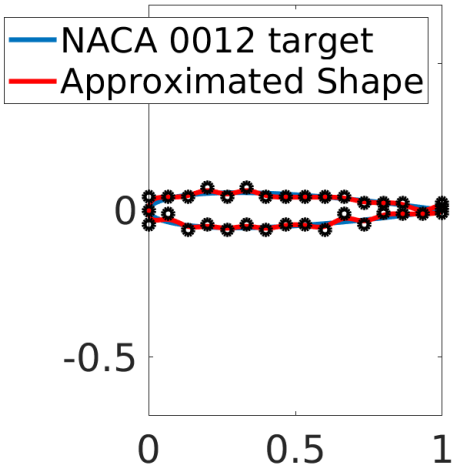
- Genetic Algorithms are typically represented by a string, this string could take many forms, such as bitsrings or real-valued numbers. What are the advantages and disadvantages of each encoding in this application? How would you convert each into 32 real-valued numbers? How could you perform crossover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

    1. Bitstring : *In Genetic Algorithms all the chromosomes are usually represented as bit strings. This encoding style remains suitable in discrete solution spaces. The reason to consider using bits instead of integers has the necessity to do with the range of data that the pool is created with. Having 32 integers means that crossover can only occur on 32 values that are taken as whole values. But having 1024 bits (32 * 32 bit integers) gives a finer granularity. And adds to the following advantages, by allowing the mutation to be bit flipping and crossover by combining the top part of one integer and the bottom part of another.*
    2. Real-valued :*If real coding is used, the algorithm uses a convex combination of the two candidates in the mating pool according to some probability pcross to produce two new candidate solutions for the next iteration. And in mutation the algorithm acts on candidate solutions, which are real coded by manipulating the value of the candidates as follows: for each value place that may be held by a candidate solution, we add the value rand([5,5]) at that value place to the actual candidate solution with probability pmut, where rand([5,5]) is a random integer in ([5,5]). This is to cover a range of mutations about the current candidate solution consistent with the place value separation in the decimal system.*

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

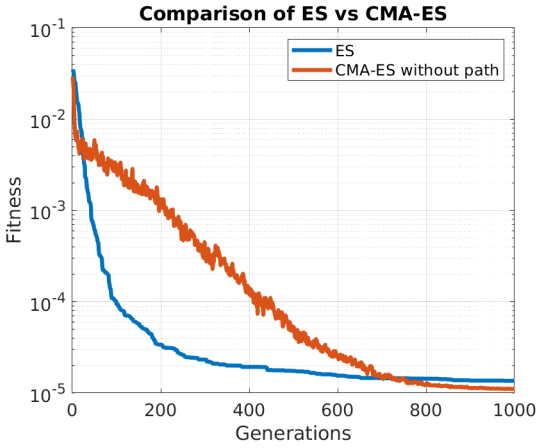
- *This is the performance of genetic algorithm with real valued numbers. We ran it for 5 iterations.*


Performance of GeneticAlgorithm
Min Fitness
Median Fitness
Fitness
Generations
0.05
0.04
0.03
0.02
0.01
0
0
20
40
60
80
100

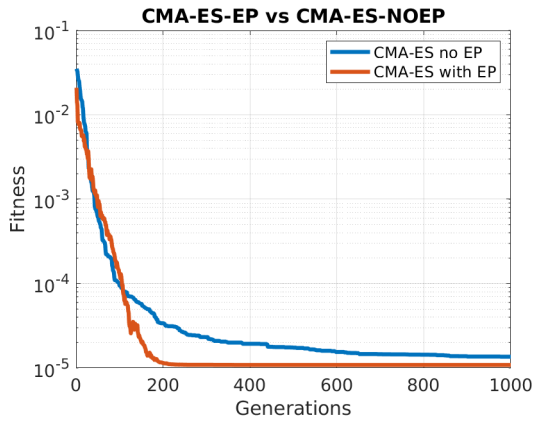

NACA 0012 target
Approximated Shape
0
-0.5
0
0.5
1

3.2 Evolution Strategies

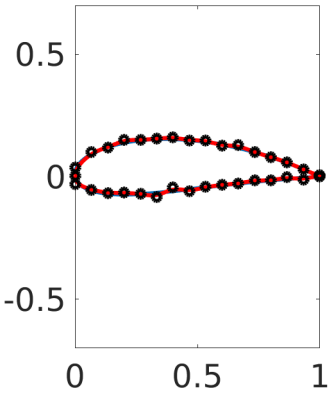
- CMA-ES is an advanced version of ES. As a first step implement a simple ES first.
  - In this ES mutation of all parameters should have equal strength which is adjusted by the 1/5th rule.
    (*every N generations change mutation strength: if* $> 1/5th$ *of mutations resulted in an better fitness (i.e. a best solution) increase mutation strength, otherwise decrease mutation strength*). Test your implementation on one of the shapes.
  - Compare to your best GA. Is one significantly better than the other? *The comparison has been made between ES and GA. The performance of ES has been relatively slower than GA.*

- Now program CMA-ES without evolution paths.
  - Compare to your ES results. Is a there a significant improvement? *Yes, there have been significant improvements. CMA-ES with no evolution paths, converges faster than ES here*
  - 

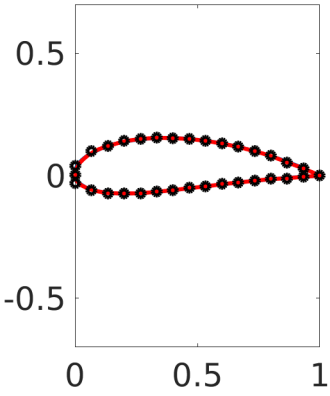
- ****Extra Credit**** Now program CMA-ES with evolution paths.
  - Compare to your previous CMA-ES results. Is a there a significant improvement? *Yes, also there have been significant improvements. CMA-ES with evolution paths, converges faster than CMA-ES with no evolution paths here.*



– *Performance of CMAES-without EP, for a single run with 200 generations*



– *Performance of CMAES-with EP, for a single run with 200 generations*



3.3 Comparisons

Produce one plot which shows the performance of each algorithm. Run each algorthm 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)
6. All the three graphs have been plotted for 3rd wing, [9,7,3,5]
7.