Genetic Evolution of a Multi-joint Virtual Creature

Mahir Gulzar, Meri Liis Treimann, Novin Shahroudi
Institute of Computer Science, University of Tartu

INTRODUCTION

We defined our own virtual creature in Unity3D to experiment a naïve evolution loosely inspired by Karl Sims virtual creatures [1]. We use a multi joint creature (so-called robotic-arm), with the aim of reaching to a target. This creature has a fixed morphology and its behavior determined by genetic algorithm trying to find the optimal joint angles which takes the end-effector (the end-point of the arm) to the target. The search space is a function of the length of arm from the base to the end-effector in initial position where are angles are zero. The complexity of the problem depends on number of joints and search space size.

METHODS

We implemented a genetic algorithm and a random search for comparison. The chromosome consists of real-value genes where each represent one joint. Each joint consists of 3 angles. Possible rotations around x, y, or z axes in 3D space are depicted in Fig. 1. We instantiate the creatures as the population and evolve them using genetic operations at each iteration. transferring elites directly to the next generation we ensure that we are going to be at least as good as previous generation. Using a combination of elites and non-elites for crossover ensures diversity in the offspring. Mutation for elites is only performed on last genes which represent closer joints to the end-effector ensures more diversity around. the target. This process is depicted in Figure-2.

RESULTS

We could achieve a solution using fixed population size of 100 in each generation. In our solution elite size to transfer directly to the next generation is set to 30. Each individual may get mutated with a chance of 1% or crossed-over. For the crossover, 40% of the elites are used as parents to generate new off-springs. Rest of the elites crossed over randomly with the remaining population.

For a given target we repeated 14 trials. The average best fitness showed in Figure 3. Also the average number of iterations required to reach to a solution was 36. Out of 14 trials 3 of them required more than 100 iterations.

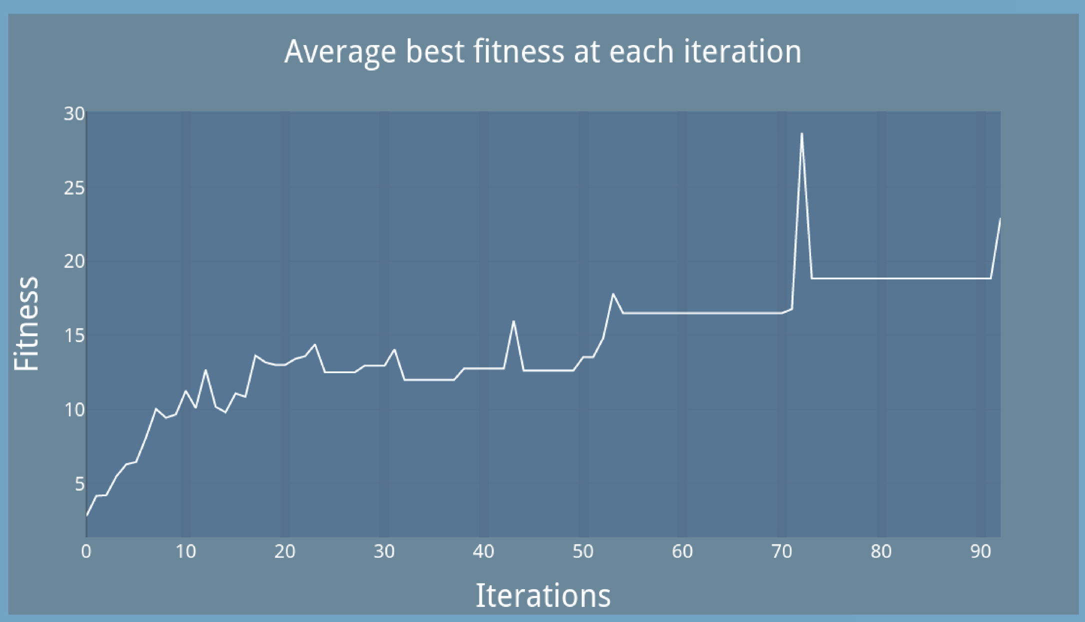

Figure 3- Degrees of Freedom for each joint

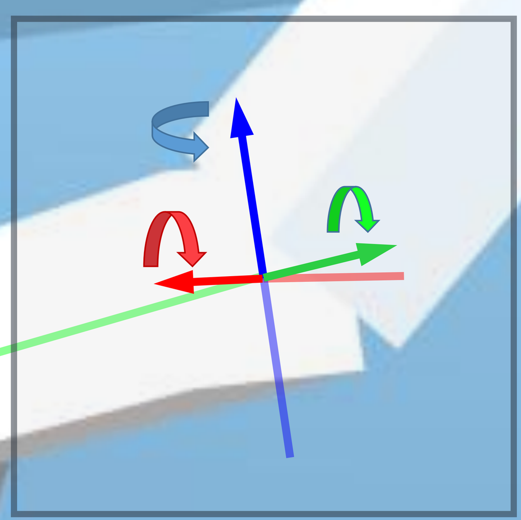

Figure 1- Degrees of Freedom for each joint

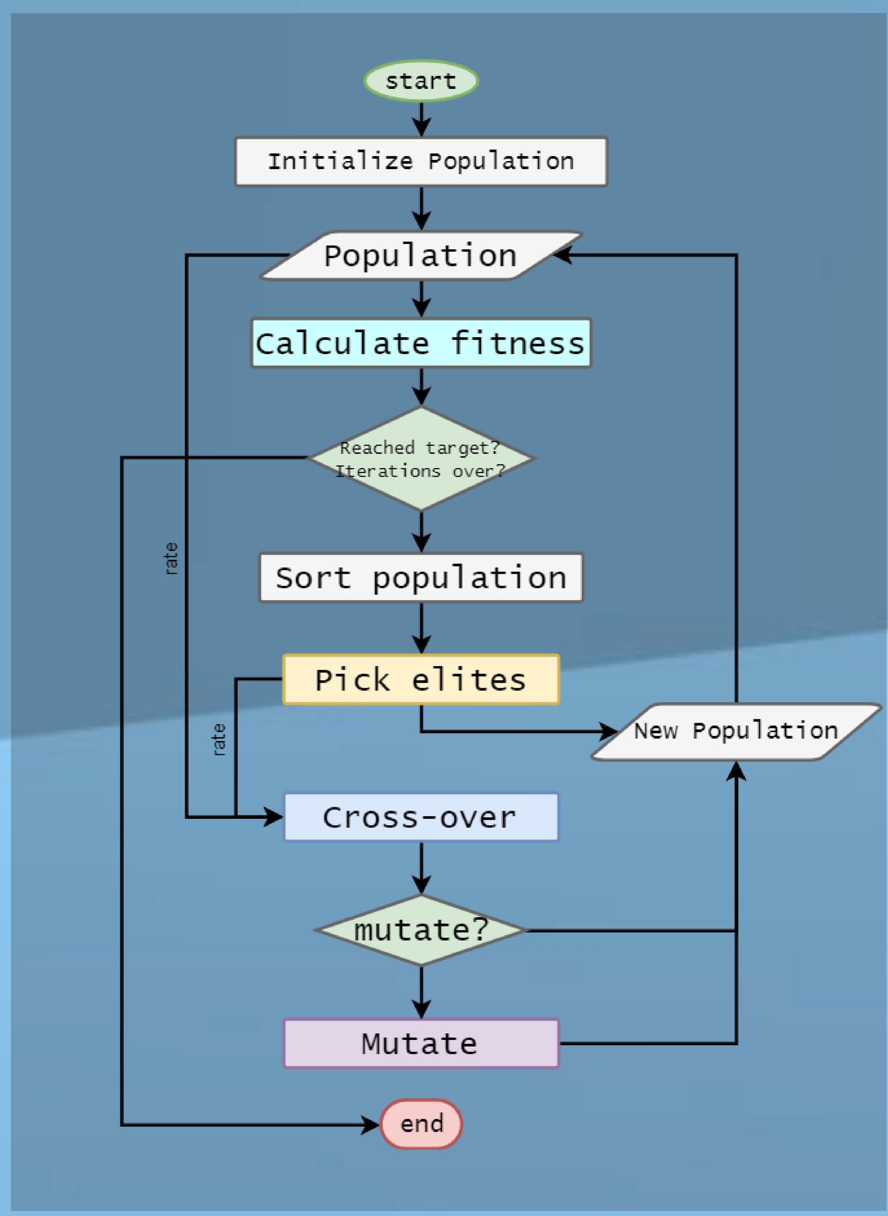

Figure 2- Flowchart of the evolution algorithm

CONCLUSION

Solvability and efficiency [2] of our approach being shown in the results. In further tests we also tried more joints for the creature and our approach showed to be extensible. Yet its efficiency requires further experiments and studies. As a future a Neural Networks maybe employed for dynamic morphology and behavior control.

References

[1] Karl Sims, Evolving virtual creatures, Computer Graphics (Siggraph '94 Proceedings), July 1994, pp.15-22.
[2] Yuval Davidor, Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization. Chapter 1

UNIVERSITY OF TARTU
Institute of Computer Science

1632