Applying Evolutionary Computation to Robotics

Adrian Thomas Schiller

Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

28 April 2014

Outline

1. Overview
2. Background
3. Research Cases
4. Simulation
5. EvolutionaryProcess
6. Results
7. Questions and Acknowledgements

The big picture

- Robots are faced with difficult problems
- Evolutionary computation is a process which can can solve difficul t problems in programming
- Since a robot interacts with the physical world, EC is slower by s everal magnitudes
- It is possible to use EC to evolve robots

Bluedrakon
`http://tr.im/pWUi`

Outline

1 Overview

2 Background
- Evolutionary Computation
- Genetic Algorithm
- “One Max” Example
- Artificial Neural Networks

3 Research Cases

4 Simulation

5 EvolutionaryProcess

Evolutionary Computation

- Evolutionary Computation (EC) is a problem solving technique which mimics natural selection

Evolutionary Computation: Requirements

- A population of candidates
- A fitness function

Evolutionary Computation: Process

- Candidates are evaluated
- The best performing candidates are selected
    - Two candidates cross-over with one another
    - Some candidates are subject to mutation
- This process repeats until the population is recreated

Genetic Algorithm

- Genetic Algorithms are a type of EC
- Candidates are represented as bit strings

Genetic Algorithm: Cross-over

- A Cross-over point is selected from two candidates. All bits are swapped beyond the point, creating two new candidates.

- Before:

$$C_i = [1, 0, 1, 0, |0, 0, 1, 0, 0, 1]$$

$$C_j = [0, 0, 1, 1, |0, 1, 1, 0, 1, 0]$$

- After:

$$[1, 0, 1, 0, 0, 1, 1, 0, 1, 0]$$

$$[0, 0, 1, 1, 0, 0, 1, 0, 0, 1]$$

Genetic Algorithm: Candidate Manipulation

- Mutation:

$$[1, 0, 1, 0, 0, 0, 1, 0, 1, 1]$$

$$[1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$$

Example: “One Max”

- Goal: Evolve an array consisting of the most ones from 20 candidate arrays with 10 bits

$$20\left\{\begin{array}{c}[1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1]\\ [0,\ 0,\ 1,\ 1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 0]\\ \ldots\\ [1,\ 0,\ 1,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0]\end{array}\right.$$

Example: "One Max"

- The fitness function $F()$ is used to calculate the fitness of each candidate
- In this case, the function counts the number of ones in the array

$$F([1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1]) = 6$$

$$F([0,\ 0,\ 1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0]) = 4$$

$$\ldots$$

$$F([1,\ 0,\ 1,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0]) = 5$$

Example: “One Max”

- The top 10 candidates are selected to create a new population
- Top candidates are randomly selected to cross-over with one another

$$C_1 = [1,1,1,0,|0,0,1,0,0,1] \qquad C_x = [1,1,1,0,0,1,1,0,1,0]$$

$$C_2 = [0,0,1,1,|0,1,1,0,1,0] \qquad C_y = [0,0,1,1,0,0,1,0,0,1]$$

Example: "One Max"

- Some created candidates may undergo mutation:

$$C_x = [1, 1, 1, 0, 0, 1, 1, 0, 1, 0]$$

$$C'_x = [1, 1, 1, 0, 1, 1, 1, 0, 1, 0]$$

Example: "One Max"

- Once the population is recreated, the process repeats until eventually a limit is reached
- In this case, the limit would be time or finding an array of all 1's

$$F([1,1,1,0,1,1,1,0,1,0]) = 7$$

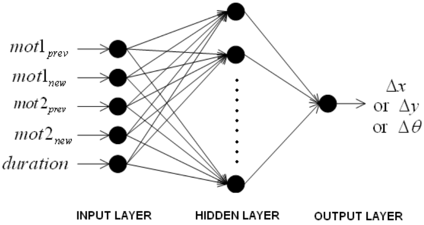
$$F([0,0,1,1,0,0,1,0,0,1]) = 4$$

$$\dots$$

$$F([1,0,1,1,0,1,0,0,1,1]) = 6$$

Artificial Neural Networks

- Artificial neural networks are a collection of nodes with weighted vertices.



Bluedrakon
`http://tr.im/pWUi`

Outline

1 Overview

2 Background

3 Research Cases
- SwimmingRobot
- WalkingRobot
- CoordRobot

4 Simulation

5 EvolutionaryProcess

6 Results

Swimming Robot

Walking Robot

Coordinate Tracking Robot

Outline

1. Overview

2. Background

3. Research Cases

4. Simulation
   - SimSwimmingRobot
   - SimWalkingRobot
   - SimCoordRobot
   - SimCoordRobot

5. EvolutionaryProcess

Swimming Robot: Simulation

Walking Robot: Simulation

Coordinate Tracking Robot: Simulation

Coordinate Tracking Robot

Outline

1 Overview

2 Background

3 Research Cases

4 Simulation

5 EvolutionaryProcess
- EPSwimmingRobot
- EPWalkingRobot: Evolutionary Process
- EPCoordRobot: Evolutionary Process

6 Results

Swimming Robot: Evolutionary Process

Walking Robot: Simulation

Coordinate Tracking Robot: Simulation

Outline

1 Overview

2 Background

3 Research Cases

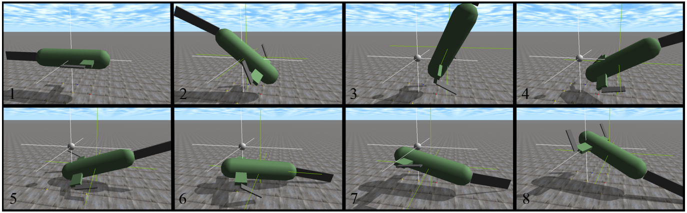
4 Simulation

5 EvolutionaryProcess

6 Results

- Station Keeping Robot
- ResultsWalkingRobot
- ResultsCoordRobot

Station Keeping Robot: Results



Moore et al.

- Each trial had a candidate which successfully maintained the position
- When the flow was coming from behind, the evolved candidate would flip end-over-head to orient itself (http://y2u.be/UufbnEGFwV4)

Results Robot: Results

Coordinate Tracking Robot: Results

Outline

1. Overview
2. Background
3. Research Cases
4. Simulation
5. EvolutionaryProcess
6. Results
7. Questions and Acknowledgements

Any Questions?
Thank you to Nic McPhee, Elena Machkasova, and Alex Jarvis