Distributed Intelligent Systems
Lab 8 Tutorial

Faezeh Rahbar

22.11.2017
Part 1: Exploring PSO

• Run PSO on two benchmark functions (Sphere and Rastrigin functions) using SwarmViz

• Observe how swarm acts when varying parameters
SwarmViz

- Make sure you only have the indicated plots marked

- Fitness landscape plot
  - A history of all particles
  - Colors indicate fitness values

- Trajectory plots
  - Movement of particles
  - Previous positions can also be plotted
SwarmViz

### Benchmark function parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness function</td>
<td>Sphere</td>
</tr>
<tr>
<td>Noise (sigma)</td>
<td>0.00</td>
</tr>
<tr>
<td>Dimension</td>
<td>24</td>
</tr>
</tbody>
</table>

### Swarm parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particles</td>
<td>30</td>
</tr>
<tr>
<td>Minimum</td>
<td>-5.12</td>
</tr>
<tr>
<td>Maximum</td>
<td>5.11</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>5.12</td>
</tr>
<tr>
<td>Inertia</td>
<td>0.60</td>
</tr>
<tr>
<td>Max iterations</td>
<td>1500</td>
</tr>
<tr>
<td>Local weight</td>
<td>2.00</td>
</tr>
<tr>
<td>Neighbor weight</td>
<td>2.00</td>
</tr>
<tr>
<td>Neighbor number</td>
<td>2</td>
</tr>
</tbody>
</table>

### PSO algorithm parameters

- Noise resistance

---

**Fitness Landscape**

- Dimension 1 vs Dimension 2

**Euclidean Distance**

- Distance vs Iteration

**Fitness (Best and average)**

- Fitness vs Iteration
Part 2 : PSO for Robotic Learning

• Obstacle avoidance
  – PSO with an Artificial Neural Network to do unsupervised robotic learning

• Design a fitness function for obstacle avoidance
  – Compare with the fitness proposed by Floreano and Mondada

• How is the performance affected by PSO parameter variations
Webots simulation

Iteration: 1
Particle: 12

10 x 20 iterations
Code Structure

**Pso_sup.c**

- **Main()**
  - Initialize world
  - Best = pso()
  - Evaluate best

- **Calc_fitness()**
  - Reposition robots randomly
  - Send candidate solutions to robots
  - Store fitness value

**Pso.c**

- **Pso()**
  - Initialize swarm
  - For each iteration
    - Move particles
    - Evaluate particles
  - Return best particle

**Obs_con.c**

- **Main()**
  - Initialize robot
  - Receive weights from supervisor
  - Run controller with weights
  - Evaluate fitness and send to supervisor
Notes

• The performances for robotic learning are printed in the console of Webots

• Please fill in the Feedback Forms on Moodle