ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Disal

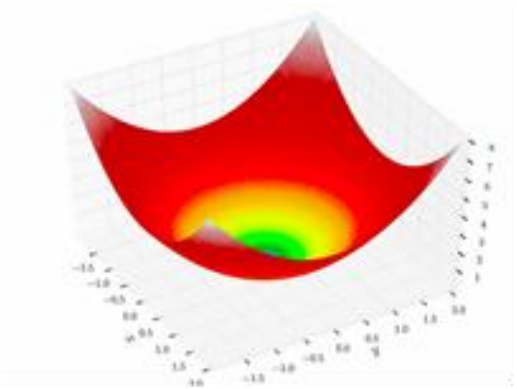
Distributed Intelligent Systems
Lab 8 Tutorial

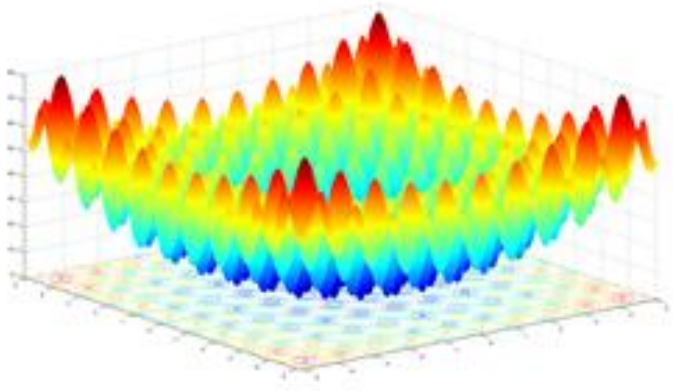
Faezeh Rahbar

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

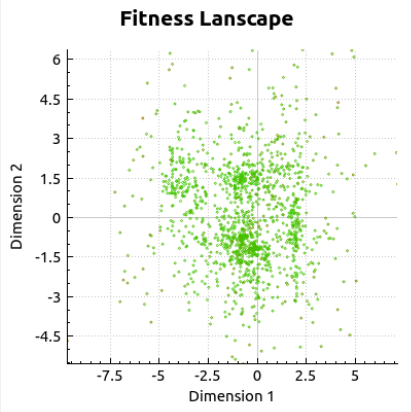
- Fitness landscape plot
  - A history of all particles
  - Colors indicate fitness values

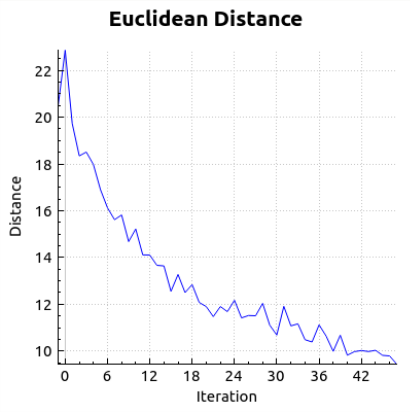
- Trajectory plots
  - Movement of particles
  - Previous positions can also be plotted

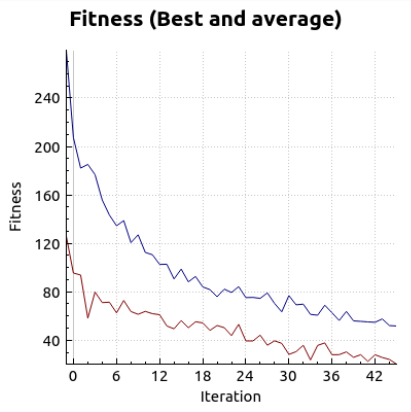
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Disal

SwarmViz







Simulation | Visualisation | Swarm | Files | Plots

Benchmark function parameters

Fitness function	Sphere
Noise (sigma)	0.00
Dimension	24

Swarm parameters

Particles	30
Minimum	-5.12
Maximum	5.11
Maximum velocity	5.12
Inertia	0.60
Max iterations	1500
Local weight	2.00
Neighbor weight	2.00
Neighbor number	2

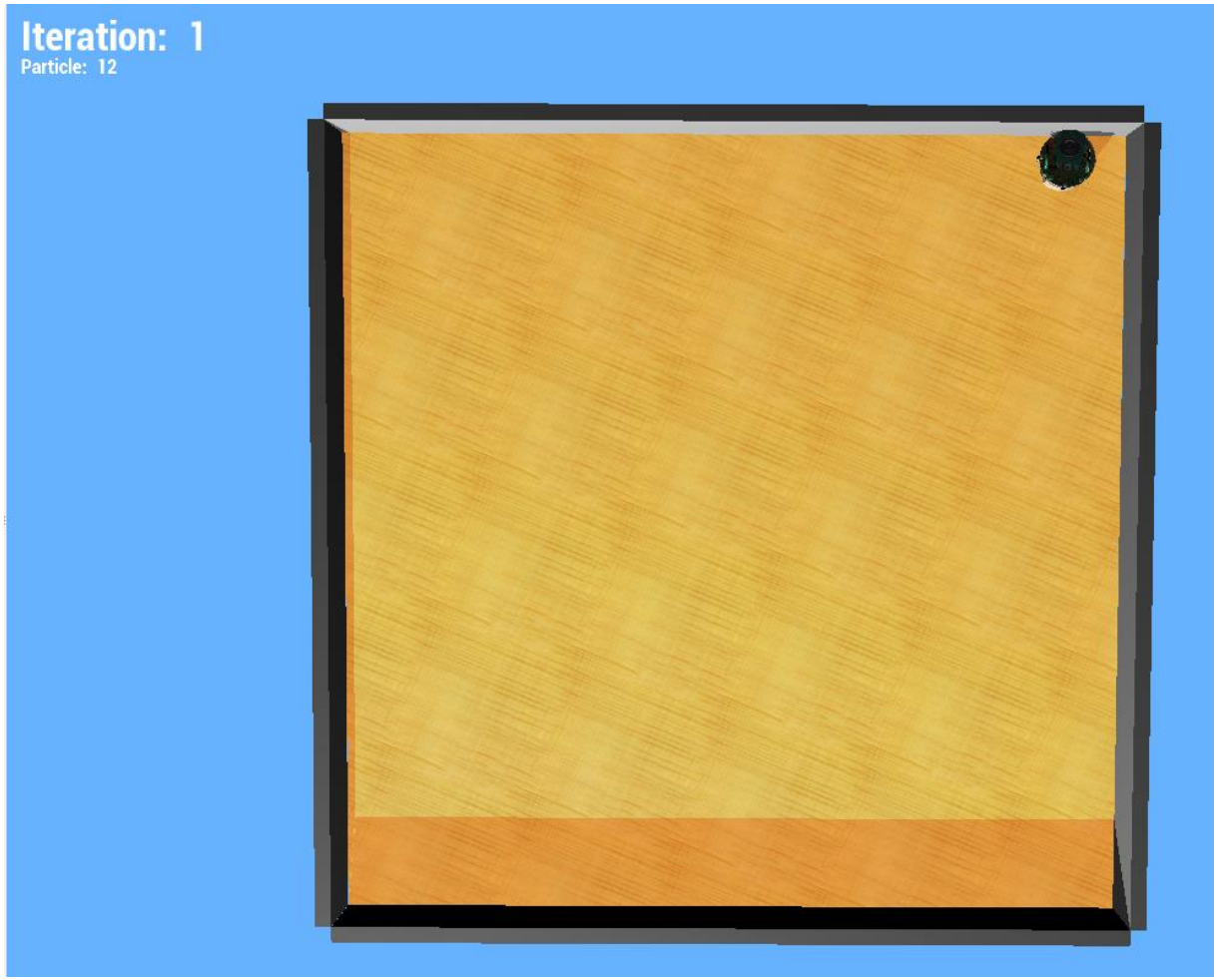
PSO algorithm parameters

☐ Noise resistance

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



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