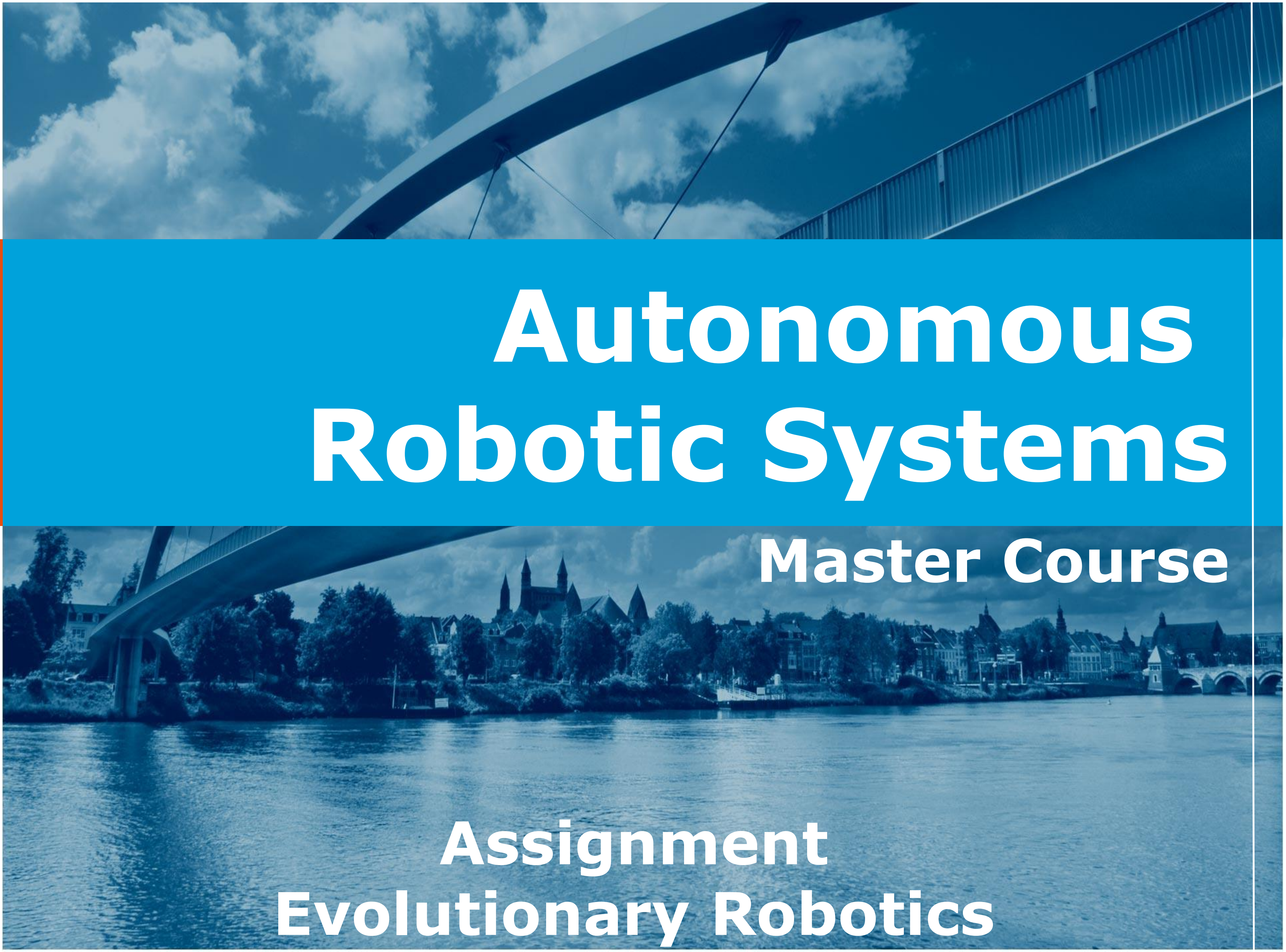


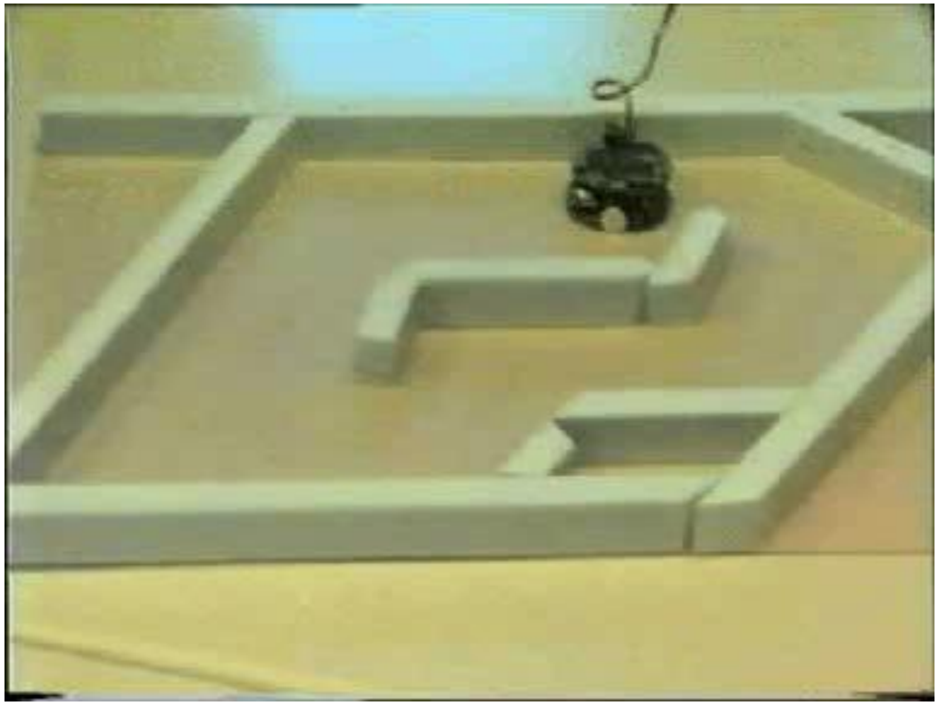
Autonomous Robotic Systems

Master Course

Assignment Evolutionary Robotics

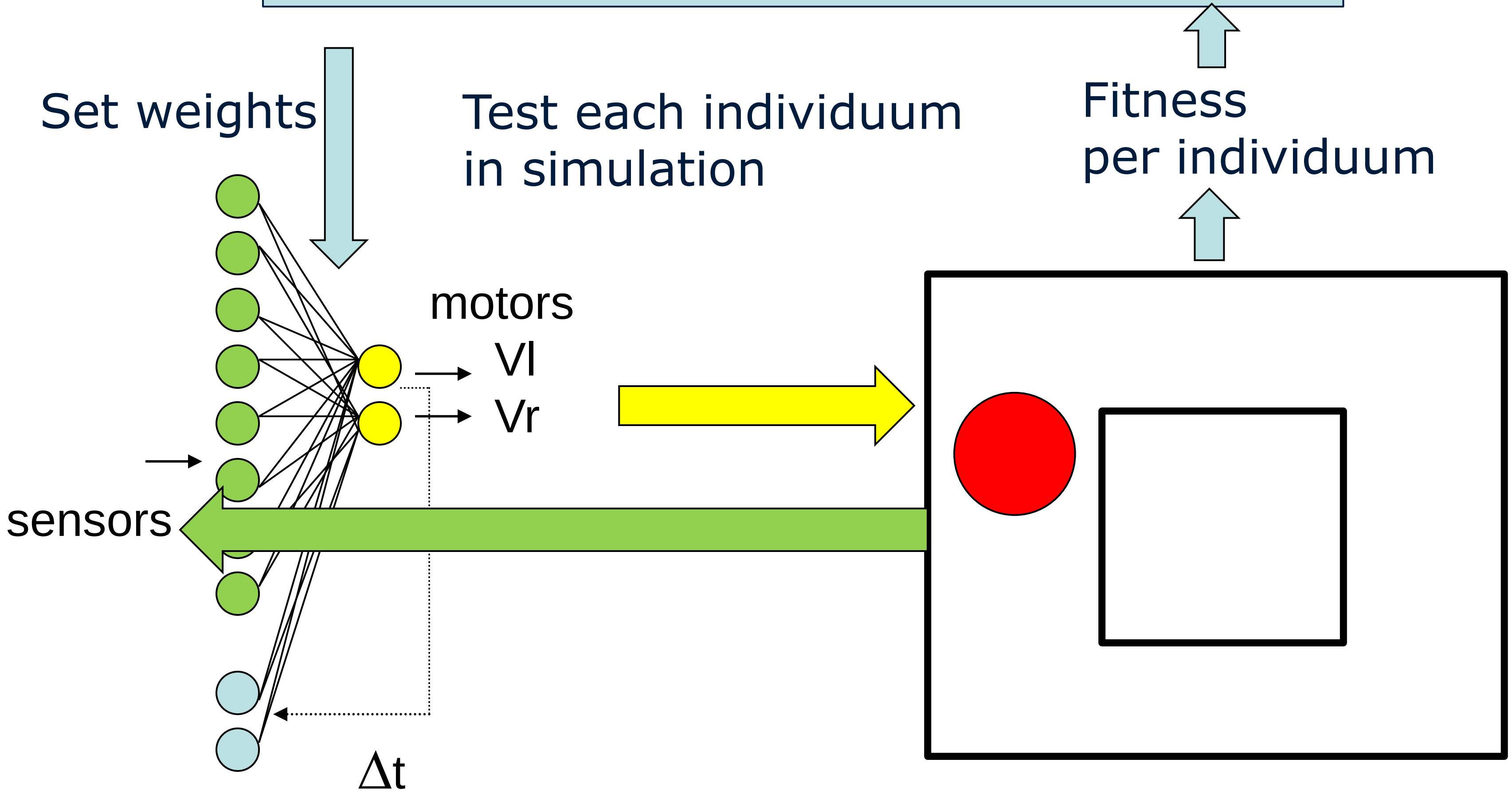
Goal: collision-free navigation of mobile robot

- Additional goal: cleaning robot: cover as much area as possible
- Use your mobile robot simulator
- Use ANN as controller
- Use EA to evolve weights of ANN (no back-propagation)



EA

Create new individuals
by selecting weights of ANN

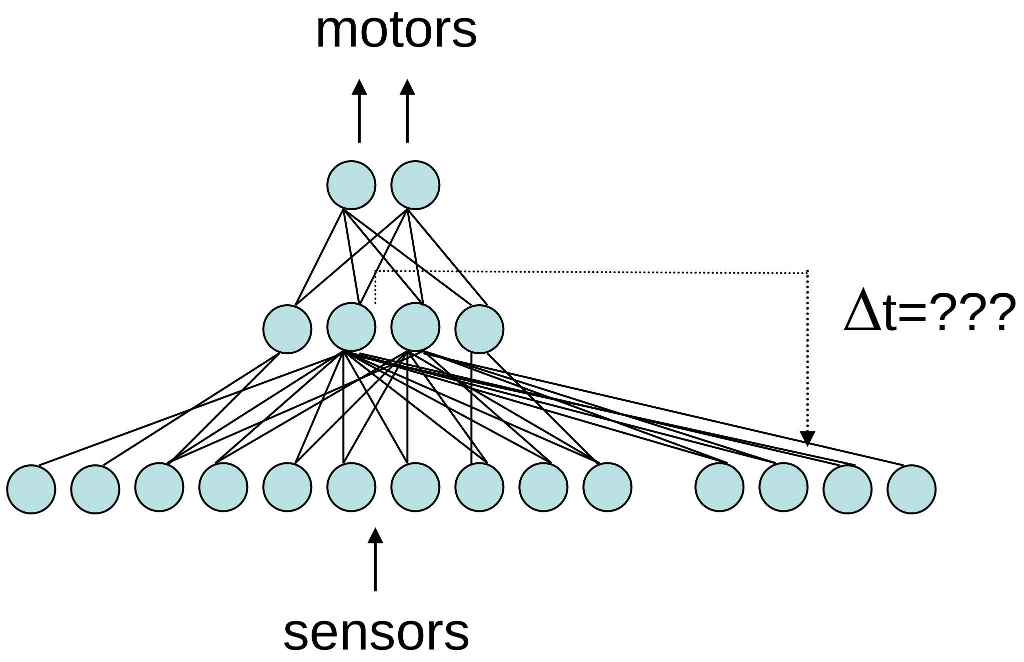


More instructions. Help your EA by shaping the fitness landscape

- Robot should clean as much area as fast as possible while avoiding collisions
- Design your own rooms
- Controller based on artificial neural network (use at least two layers with recurrent nodes)
- Robot with two wheels, ANN with two outputs – each output controls speed of one wheel
- 12 infrared distance sensors (30° distance) as input to ANN (it makes sense to use a max distance parameter)

ANN

- Need at least two layers.
- Use feedback to create memory
- Play with Δt (will depend on time step)



How to encode fitness criteria?

- Collision-free?
- Clean as much area as possible?
- How to weight both criteria?
- What about experiment?
  - Fix time for each experiment?
- Avoid that too many individuals receive zero fitness. Otherwise your EA gets stuck right from the beginning.

→ Requires some testing, but think first!!!

Make your life easier by shaping your fitness landscape/evaluation

- Cover as much area as possible: simulate dust, use removed dust as fitness

→Avoids robots that do not move enough

→Automatically covers speed criterion (if time for experiment is fixed)

→Motivates robots to move closer to walls (without collision)

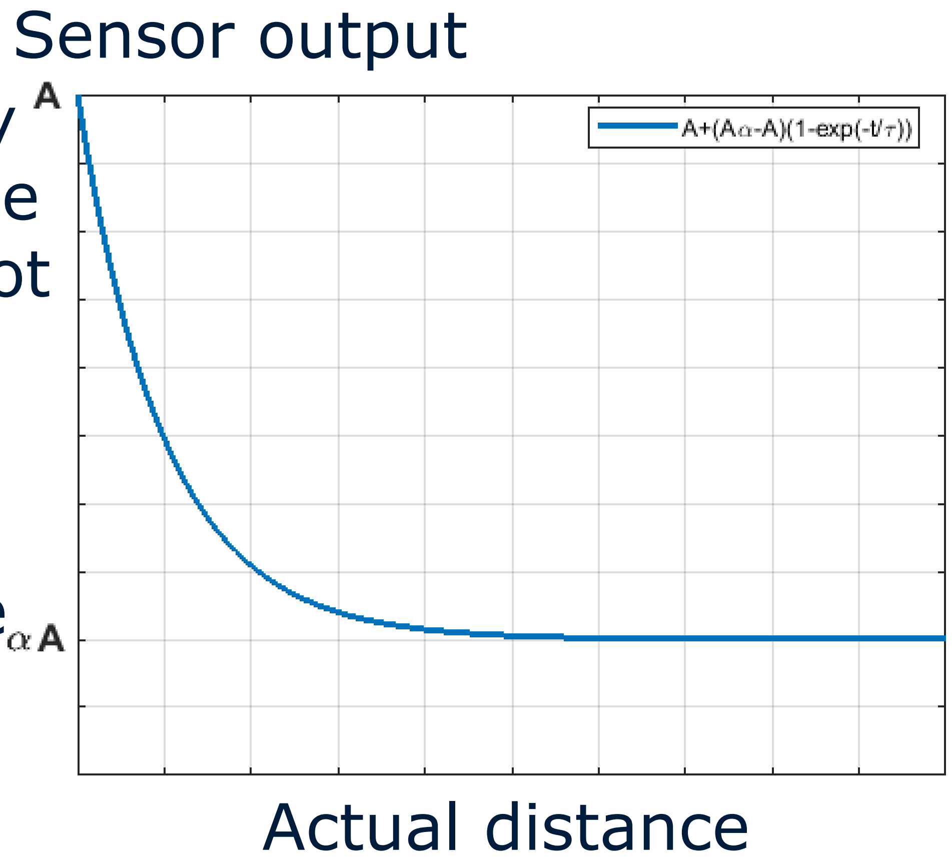
Make your life easier by shaping your fitness landscape/evaluation

- Limit range of sensors

→ Makes sure that far away obstacles do not influence the behaviour of the robot

- Shape feedback of sensors:

→ Allows ANN to "understand" importance of distances



Advice on test rooms

- To achieve robustness you want to evolve your robot using different training rooms
- Just because you think that a room is simple, it might not be for a robot (see narrow passages, open spaces etc.)
- Test your final robots on these training rooms plus on rooms that they have not seen before to allow for a strong discussion

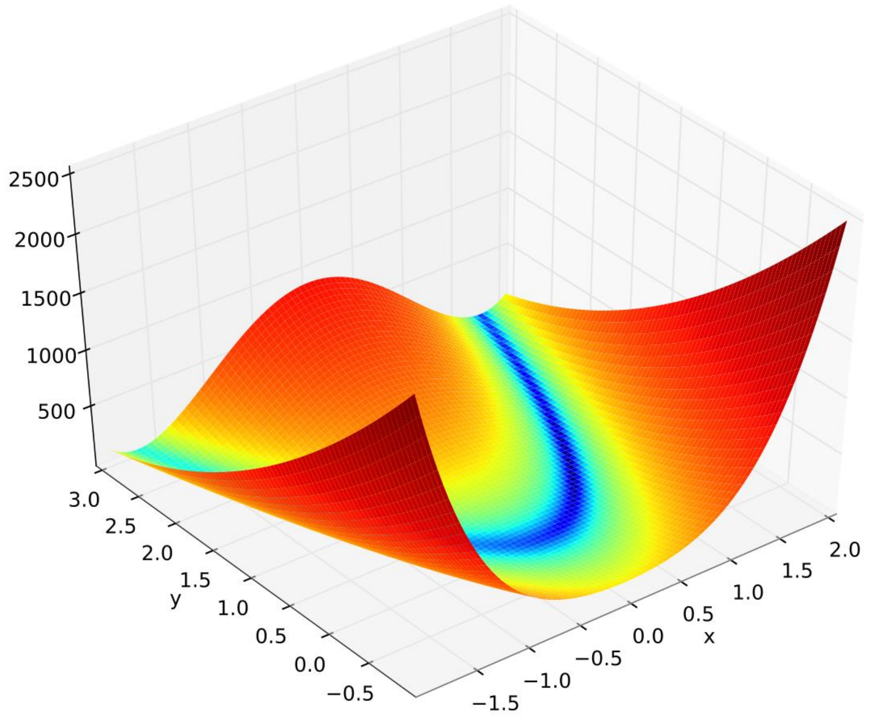
Hand in

in one work week: EA: Intermediate code & video demonstration (mp4 only!, 4 minutes max, 150MB max) EA on benchmark functions (same demos as for PSO, show updates of individuals)

in two work weeks: ER:

- Final documented code (Python, C++, C, Java, Matlab)
  - Make sure that each group member codes something, add names to code (who did what?)
  - Upload zip archive
- Final video demonstration (mp4 only!, 5-6 minutes max, 150MB max) of evolved behaviour in class

Benchmark function: Rosenbrock

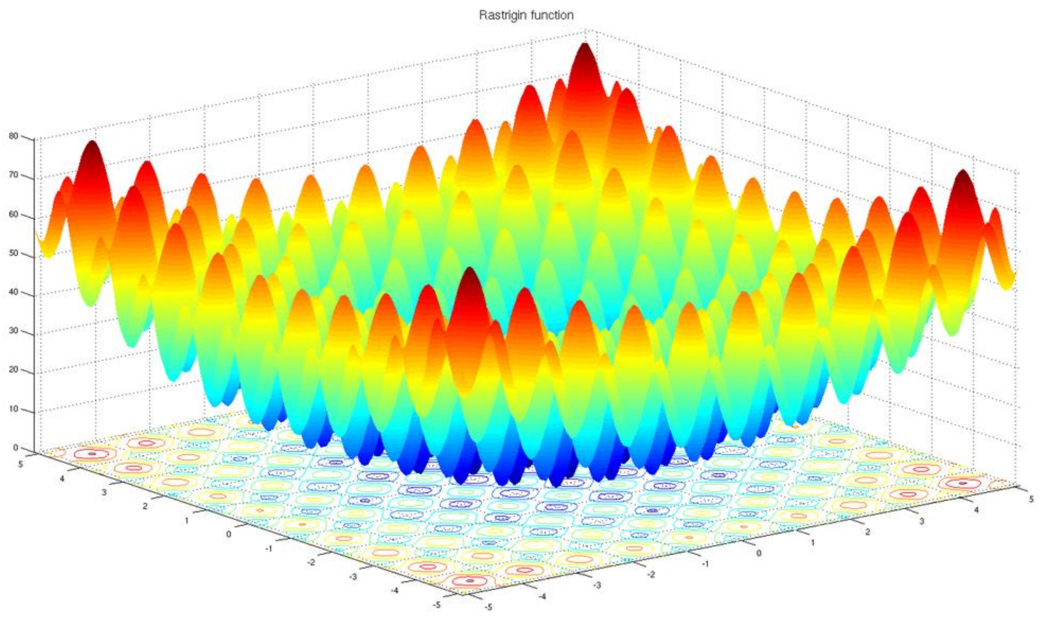


Rosenbrock function with two variables

$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

- Deep valley with known minimum minimum at (a, a^2)
- We use a=0
- Easy to find valley
- Difficult to find global minimum
- Also defined for multiple dimensions (see Wikipedia)

Benchmark function: Rastrigin



Rastrigin function with two variables

$$f_2(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10cos(2\pi x_i) \right)$$

- Many local minima
- Known global minimum at **x=0**

Documentation of EA/ER experiments
(be precise! be consise!)

Add overview slides for experiments into video documentation

- Control (ER: drawing of ANN, exact number of nodes, type of neurons)
- Morphology (ER: here simulation, state which simulator you are using)
- Genetic representation (EA/ER: exact genome)
- Population size (EA/ER: exact number)
- Initialization (EA/ER: how do you initialize genomes?) -> exact distributions
- Exact fitness function (EA/ER)
- Selection and reproduction (EA/ER: be precise)
- Crossover & mutation (EA/ER: be precise)
- Stop criterion (EA/ER: be precise)

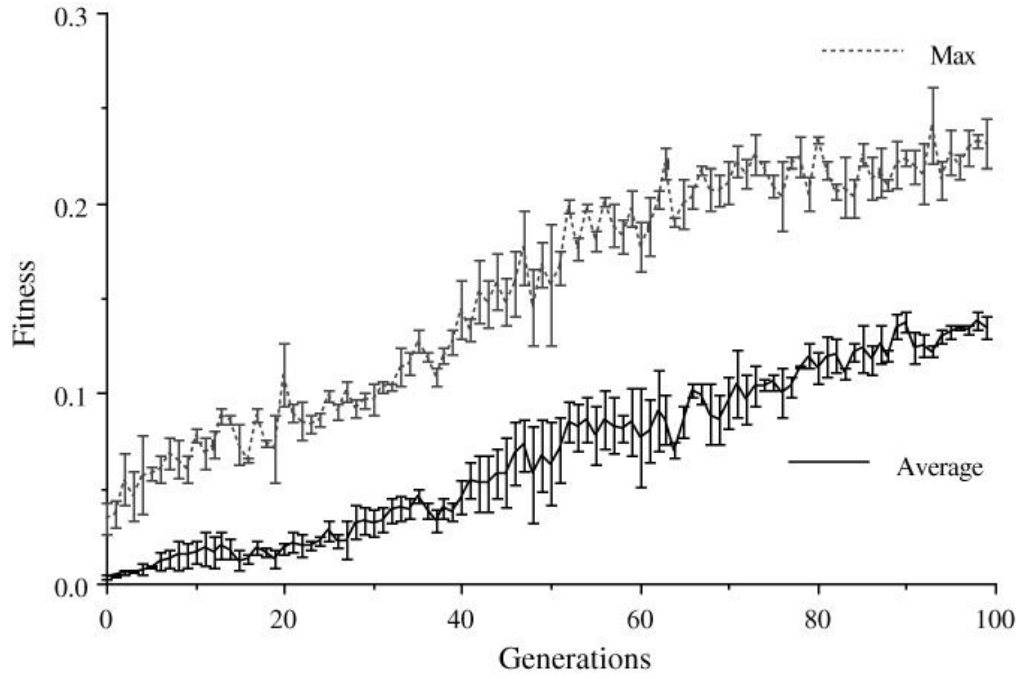
Name methods & provide precise parameters (no need to explain standard methods) e.g. tournament selection, k = 5

Video demonstrations

- Explain fitness evaluation, experiments (concentrate on successful ones) & results
- Required graphs:
  - Show evolution of fitness (max + average)
  - Show intermediated & final evolved trajectory of robot
- Discuss results (**explain what is happening and why it is happening**)
- Do not forget to put all student names
- Make sure that all students can be heard in video

Monitoring Performance

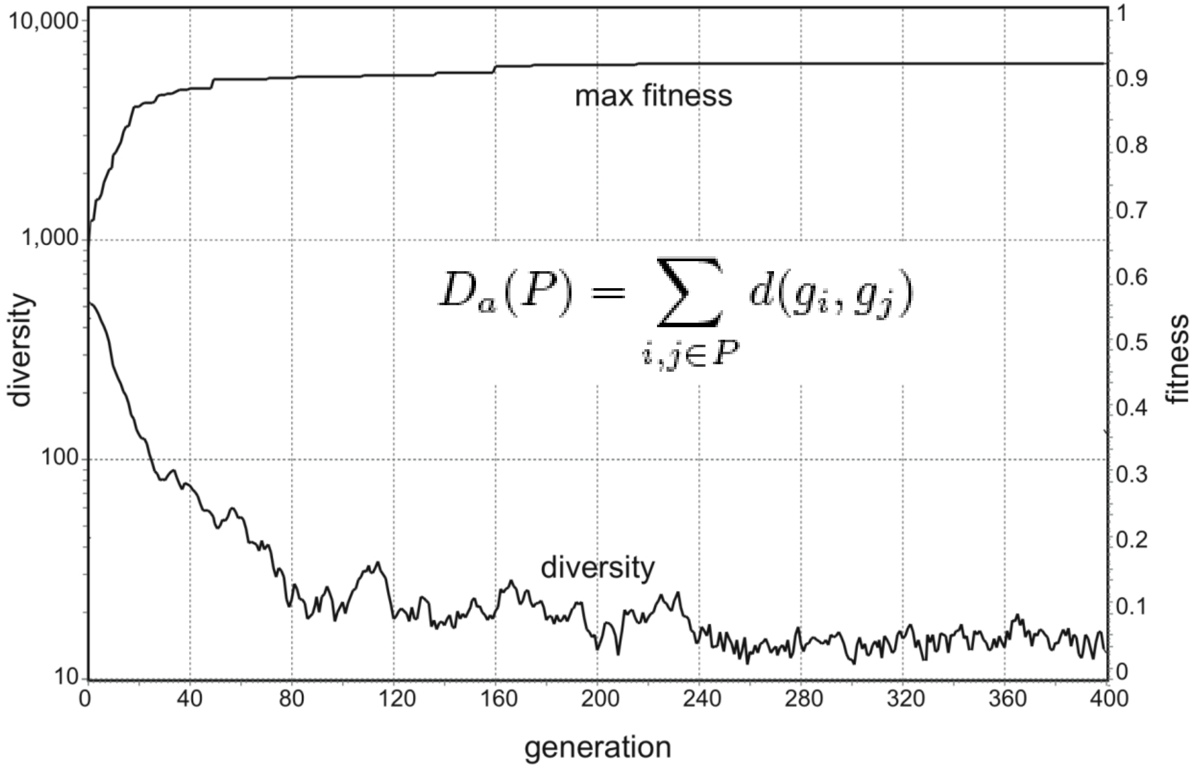
Track best and population average fitness of each generation
Multiple runs are necessary: plot average data and standard error



- Fitness graphs are meaningful only if the problem is stationary!
- Stagnation of fitness function may mean best solution found **or premature convergence**

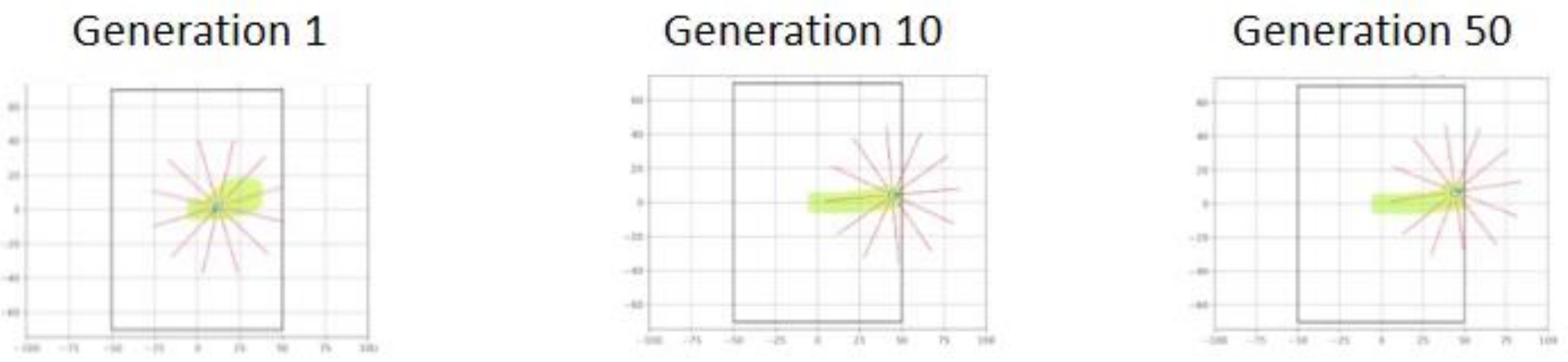
Measuring Diversity

Diversity tells whether the population has potential for further evolution
Measures of diversity depend on genetic representation
E.g., for binary and real valued, use sum of Euclidean or Hamming distances



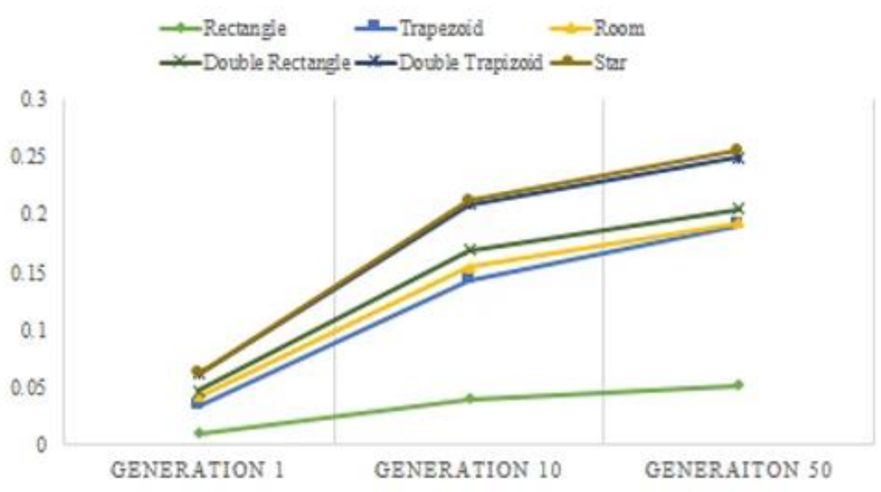
$$D_a(P) = \sum_{i,j \in P} d(g_i, g_j)$$

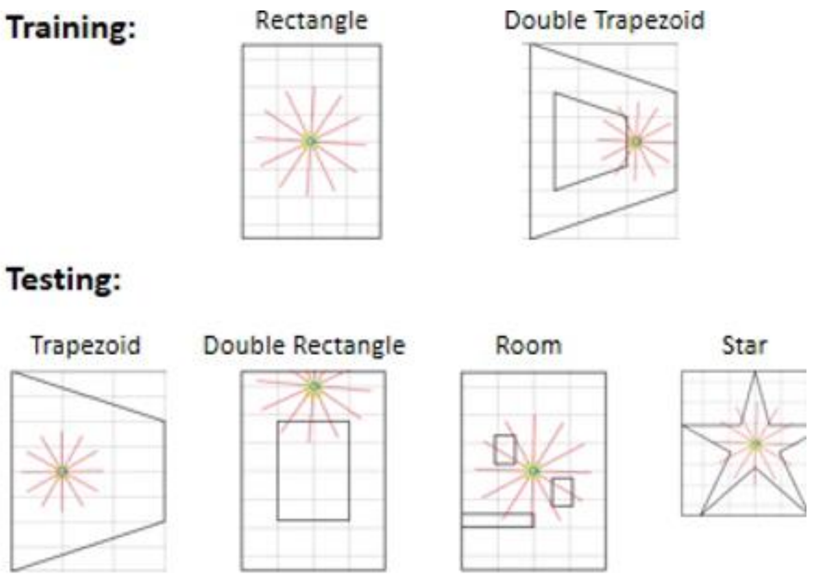
Possible final result



Show intermediate & final results

Possible final result





Test different mazes

Plagiarism

- This is a group assignment
- Help other members of your group
- Do not copy and hand in code or reports from other groups (or other parties)
- **You must write code for EA, ANN, evaluations yourself (no external libraries allowed)!!!**
- You **can** use the simulator of another group with a proper motion/sensor/collision model
- Write your own software

Write simple software

- No need to use (a lot of) objects
- Use functions
- Do not distribute code over too many files
- Avoid complex constructions and data structures