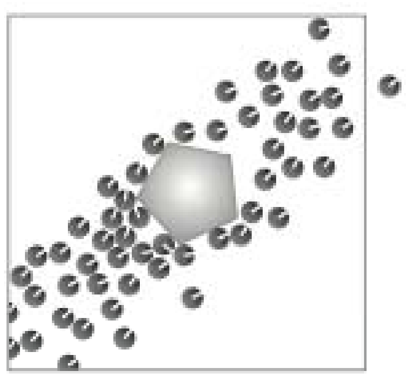
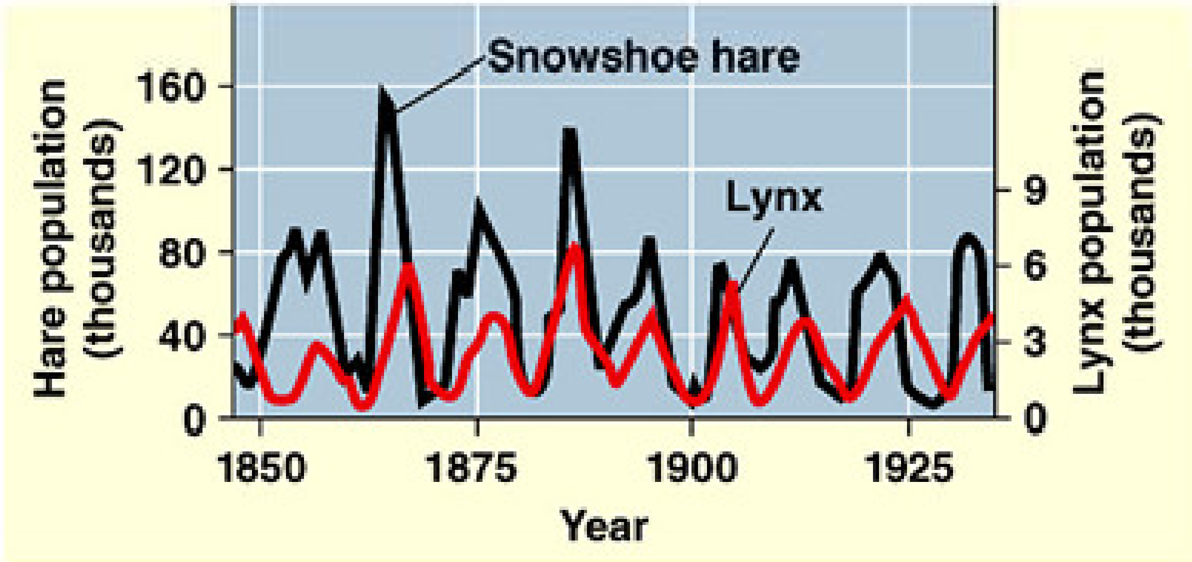
Competitive and Cooperative Co-Evolution




Snowshoe hare
Lynx
Hare population (thousands)
Lynx population (thousands)
160
120
80
40
0
9
6
3
0
1850
1875
1900
1925
Year

Competitive Coevolution

Competitive Co-Evolution is a situation where two different species co-evolve against each other. Typical examples are:

- Prey-Predator
- Host-Parasite

Fitness of each species depends on fitness of opponent species.

Potential advantages of Competitive Co-evolution:

– It may increase adaptivity by producing an evolutionary *arms race* [Dawkins & Krebs, 1979]

– More complex solutions may *incrementally* emerge as each population tries to win over the opponent

– It may be a solution to the *boostrap* problem

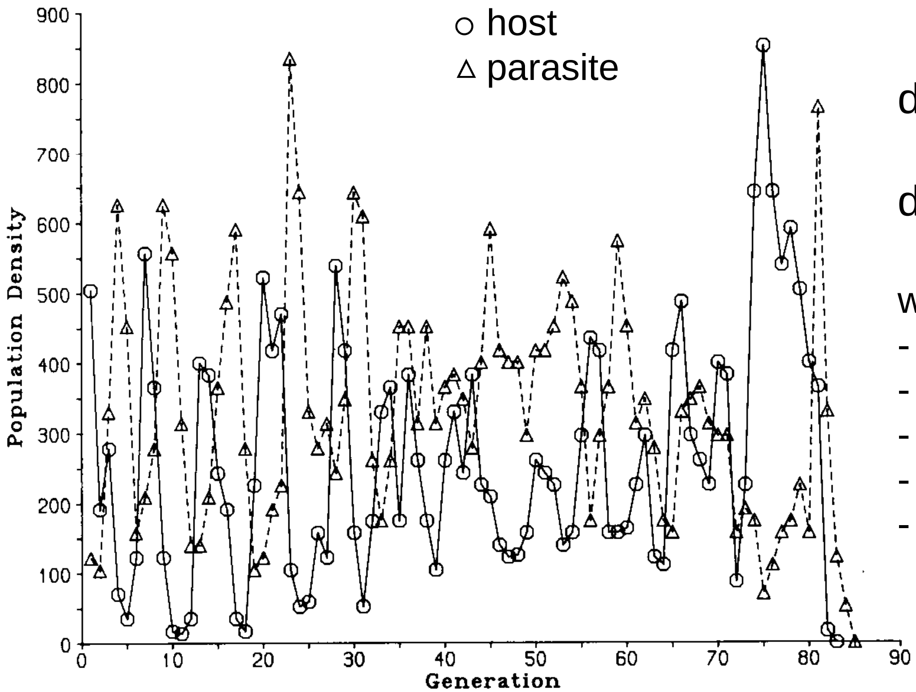
– Human-designed fitness function plays a less important role (= autonomous systems)

– Continuously *changing fitness landscape* may help to prevent stagnation in local minima [Hillis, 1990]

Formal model

Formal models of competitive co-evolution are based on the Lotka-Volterra set of differential equations describing variation in <u>population size</u>.

Notice that in biology what matters is variation in population size, not behavioral performance, which is difficult to define and



$$dN1/dt = N1\,(r1 - b1N2)$$

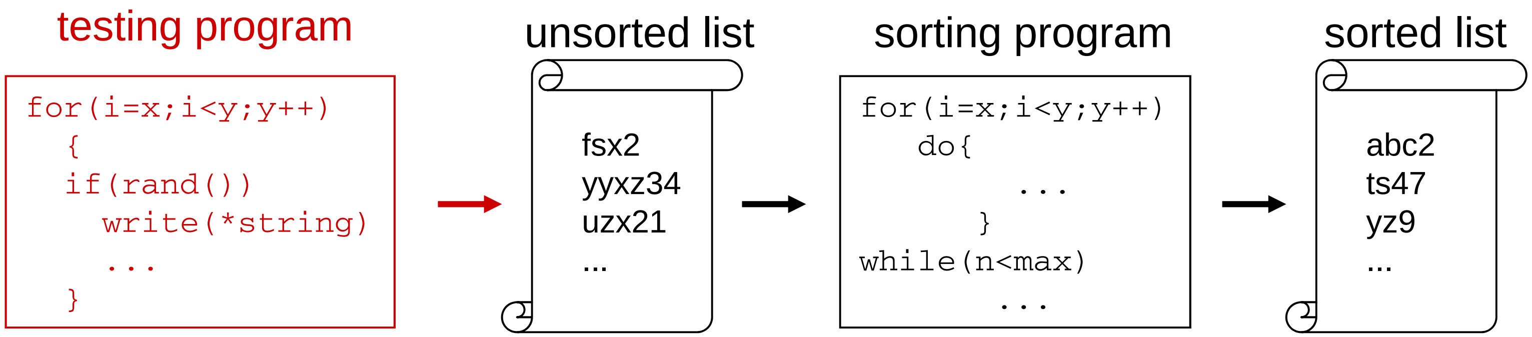
$$dN2/dt = N2\,(-r2 + b2N1)$$

where:
- N1, N2 are the two populations
- r1 is increment rate of prey without predators
- r2 is death rate of predators without prey
- b1 is death rate of prey caused by predators
- b2 is ability of predators to catch prey

Computational model

Formal models assume that behavioral performances of the two species remain constant across generations and therefore cannot be used to predict under what circumstances competitive co-evolution can generate increasingly complex (= higher fitness) individuals.

Hillis (1990) showed that co-evolution can produce more efficient <u>sorting programs</u> than evolution alone (or hand design).

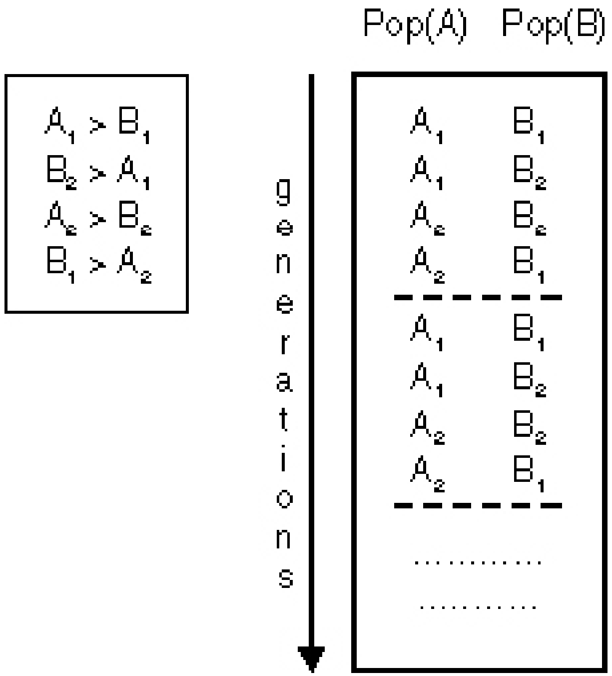


f(sorting) = quality

f(testing) = 1 - quality

Complication: Strategy recycling

The same set of solutions may be discovered over and over again across generations. After some initial progress, this cycling behavior may stagnate in relatively simple solutions.



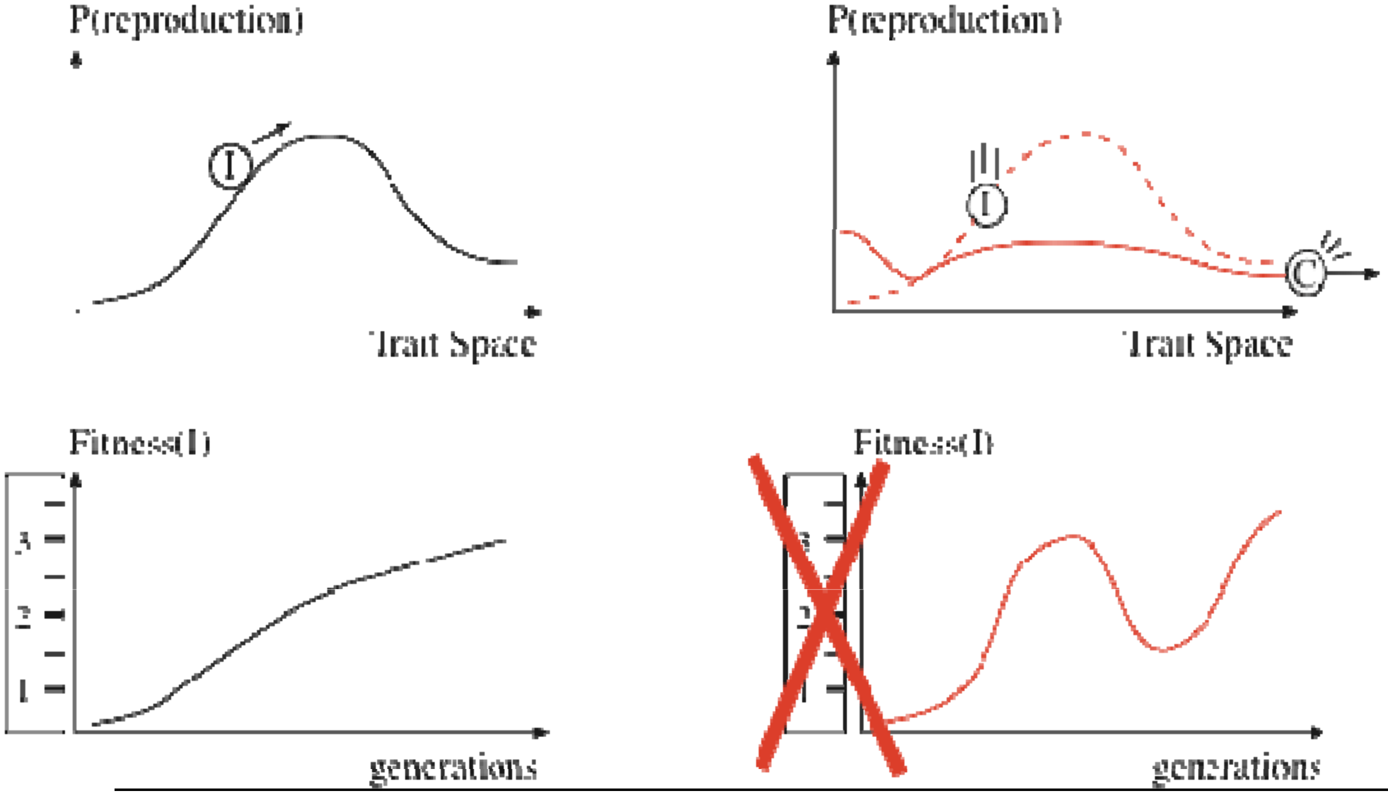
Possible causes of recycling:

- Lack of « generational memory »
- Restricted possibility for variation
- Small genetic diversity

Complication: Dynamic fitness landscape

Whereas in single-species evolution the fitness landscape is static and fitness is a monotonic function of progress, in competitive co-evolution the fitness landscape can be modified by the competitor and fitness function is no longer an indicator of progress.

single evolution — competitive co-evolution

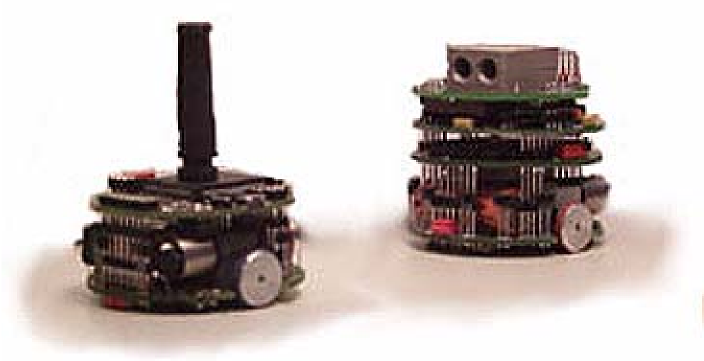
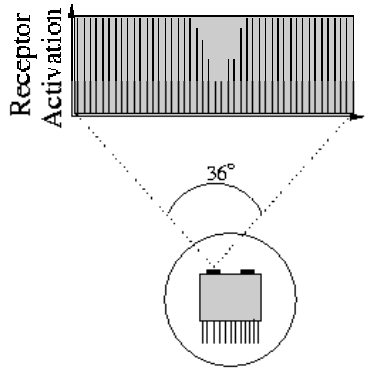


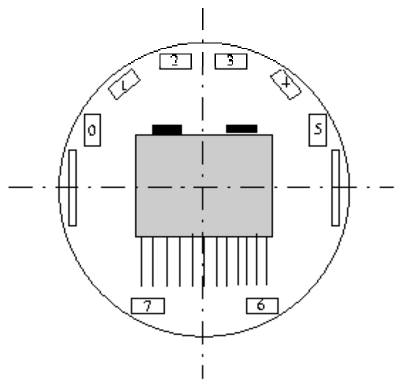
Investigation with robots

Let us consider the case of two co-evolutionary robots, a predator and a prey, that evolve in competition with each other. Questions:

a) can we evolve functional controllers with simple fitness functions?
b) what are the emerging dynamics?
c) do we observe incremental progress?
d) are co-evolved solutions better than evolved solutions?

<u>Goal</u> = Predator must catch the prey, prey must avoid predator
<u>Prey</u> = proximity sensors only, twice as fast as predator
<u>Predator</u> = proximity + vision, but half max speed of prey

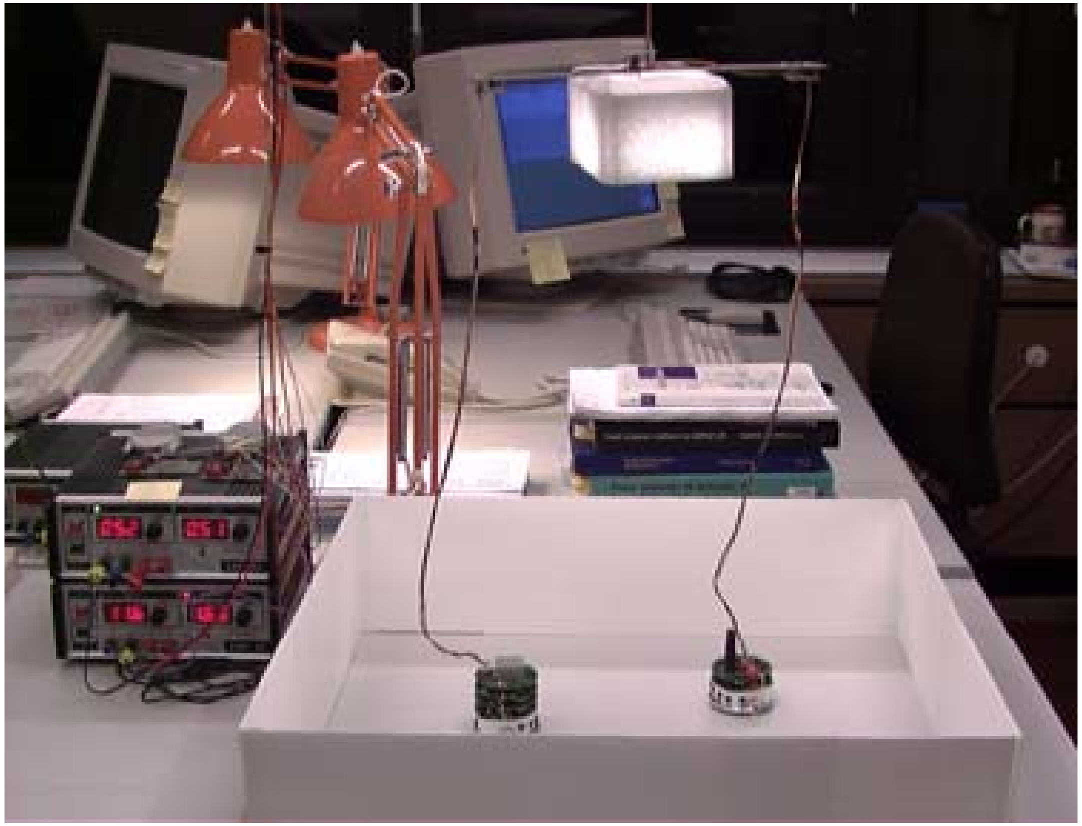





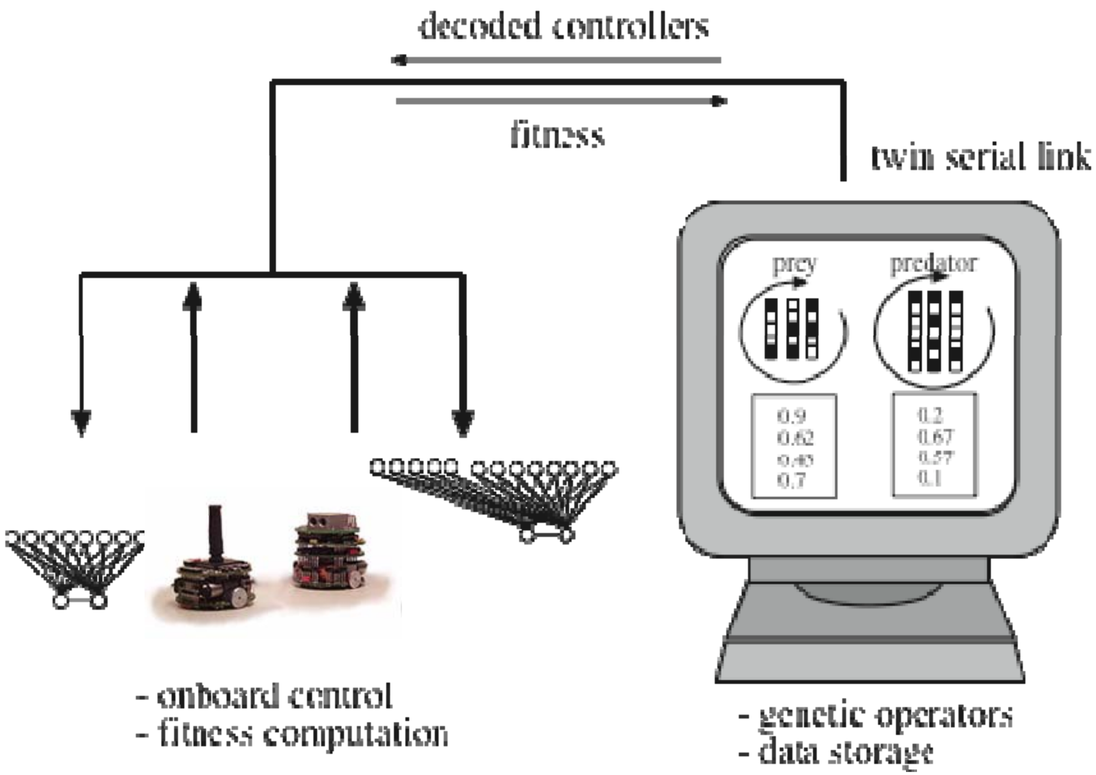
Experimental setup

The two robots are positioned in a white arena. Predator and prey are tested in tournaments lasting 2 minutes. Robots are equipped with contact sensors.

Fitness prey = TimeToContact   Fitness predator = 1-TimeToContact



Co-evolutionary algorithm

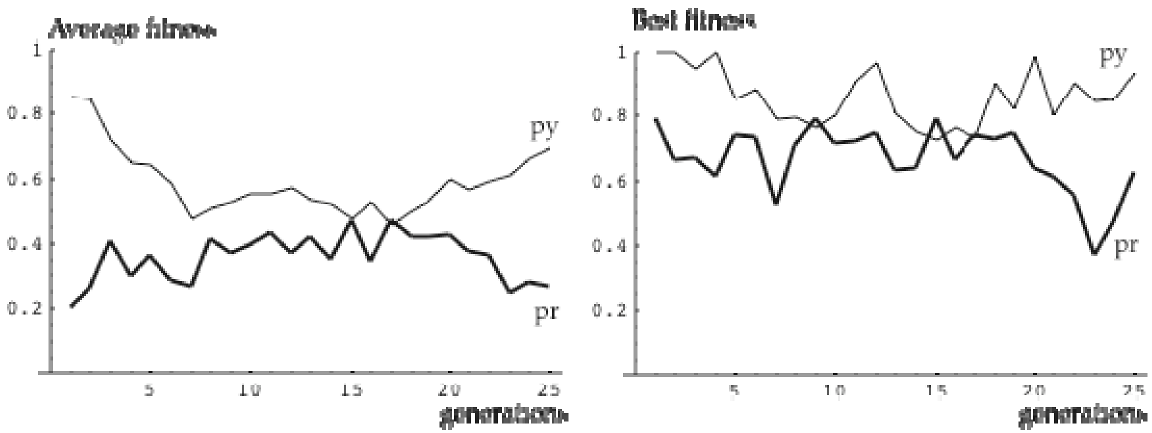


Two populations, one for the prey and one for the predator, are maintained in the computer. Each individual of one population is tested against the best opponents of the previous 5 generations.

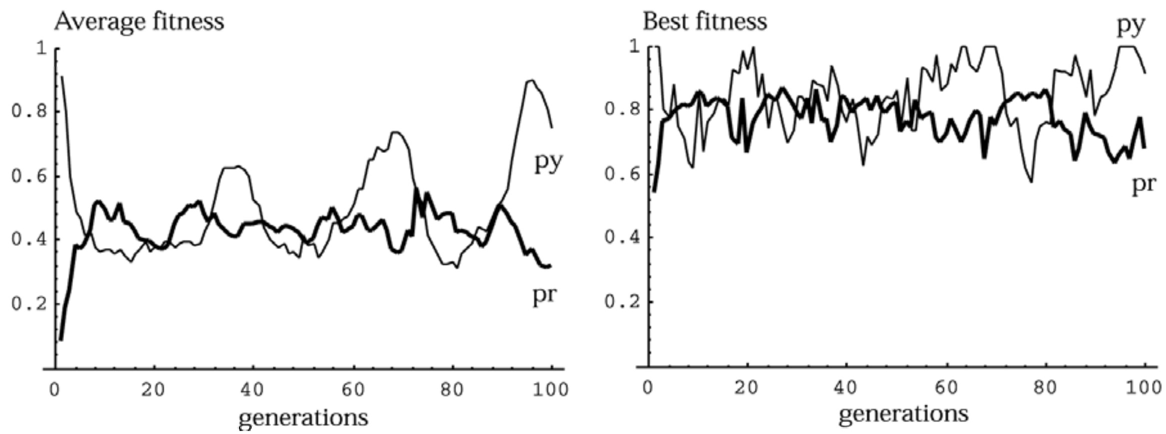
Experimental results

As expected, average and best fitness graph display oscillations.

with real robots



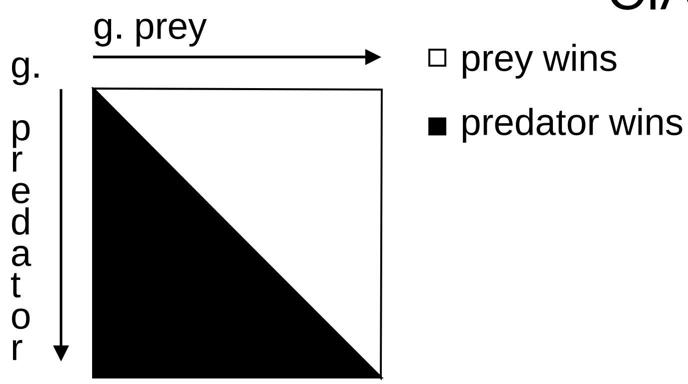
with simulated robots



Measures of progress

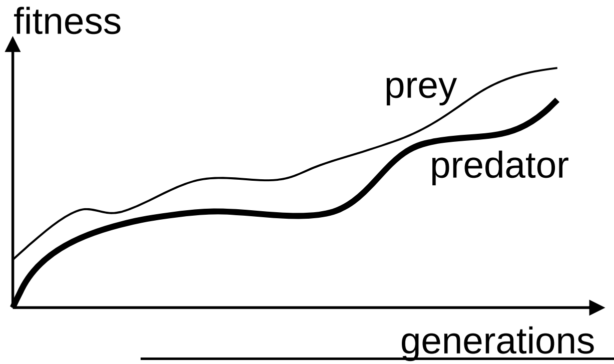
Progress can be measured by testing <u>evolved</u> individuals against all best opponents of previous generations. There are two ways of doing so.

CIAO graphs [Cliff & Miller, 1997]



These graphs represent the outcome of tournaments of the Current Individual vs. Ancestral Opponent across generations. Ideal continuous progress would be indicated by lower diagonal portion in black and upper diagonal portion in white.

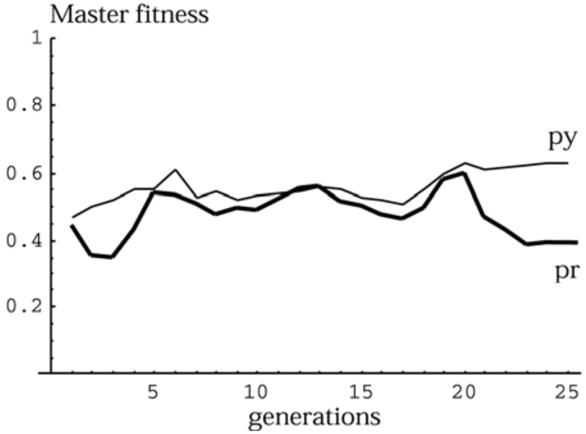
MASTER tournaments [Floreano & Nolfi, 1997a]



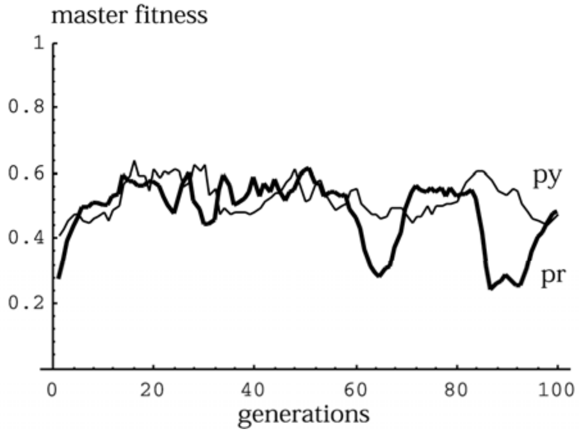
These graphs plot the average outcome of tournaments of the current individual against <u>all previous best</u> opponents. Ideal continuous progree would be indicated by continuous growth.

Limited observed progress

with real robots

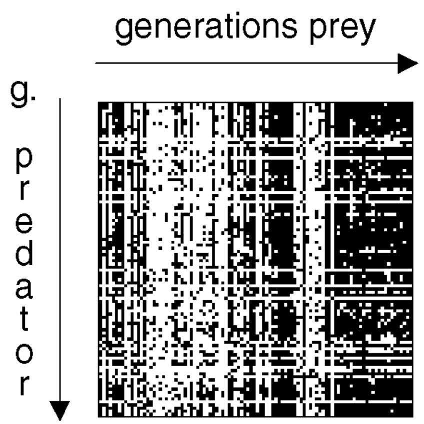


with simulated robots



Progress analysis of co-evolved robots using Master Tournament technique shows that there is some progress only during the initial 20 generations. After that, the graphs are flat or even decreasing. In other words, individuals born after 50 generations may be defeated by individuals that were born 30 generations earlier.

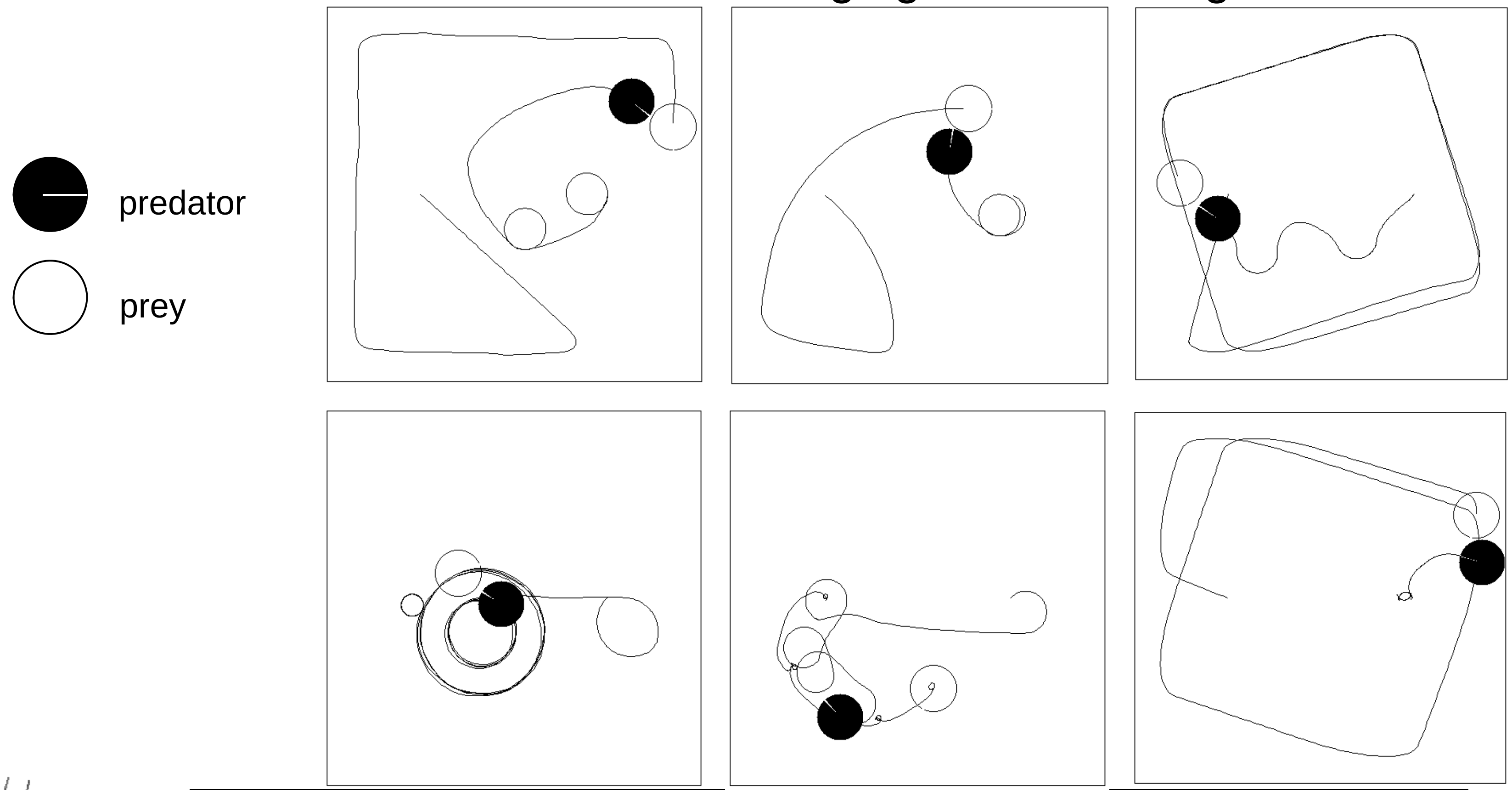
These data indicate that co-evolution may have developed into re-cycling dynamics after 20 generations.

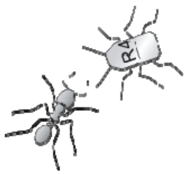


CIAO data are even less capable of revealing progress.

Evolved strategies

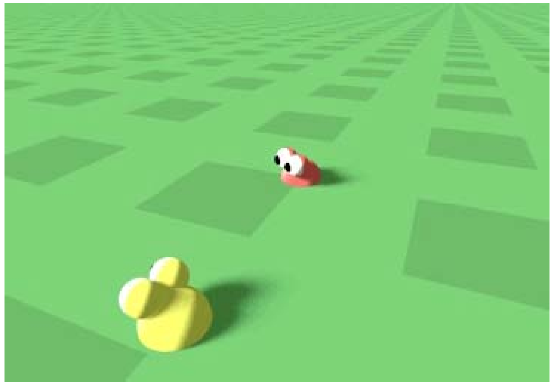
Despite lack of progress measured against previous opponents, co-evolved individuals display highly-adapted strategies against their opponents and a large variations of behaviors.
Each tournament shows individuals belonging to the same generation.





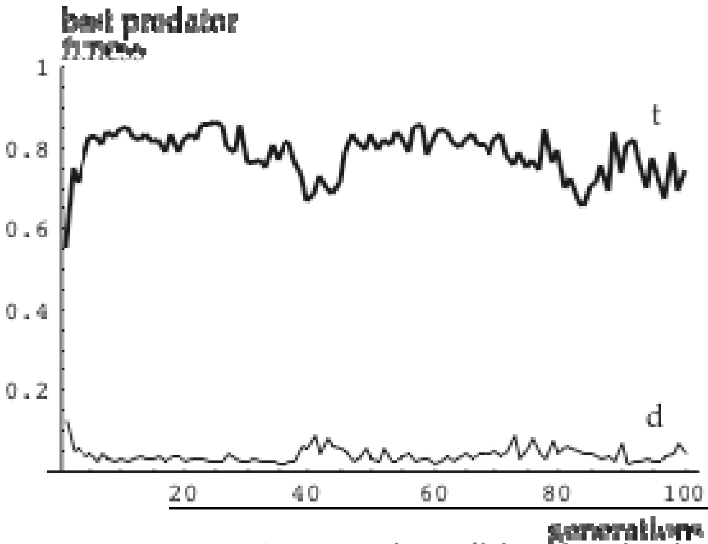
The influence of selection criteria

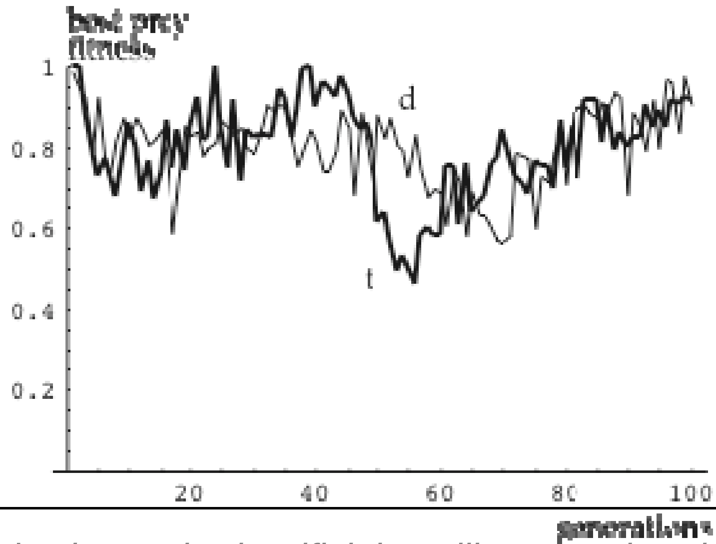
predator from g. 999
vs. prey from g. 200



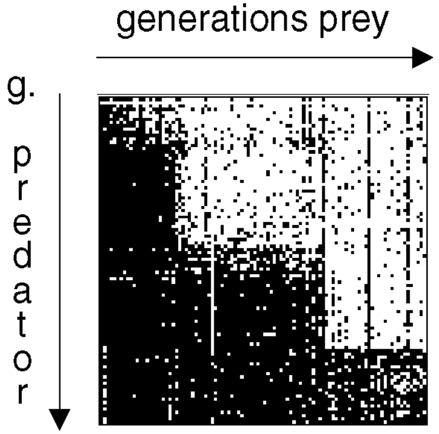
Miller and Cliff [1997] carried out a similar experiment in simulation, but used distance, instead of time, as fitness function. In Fitness Space distance is an external component whereas time is an internal one. It was difficult to evolve efficient chasing-escaping strategies.

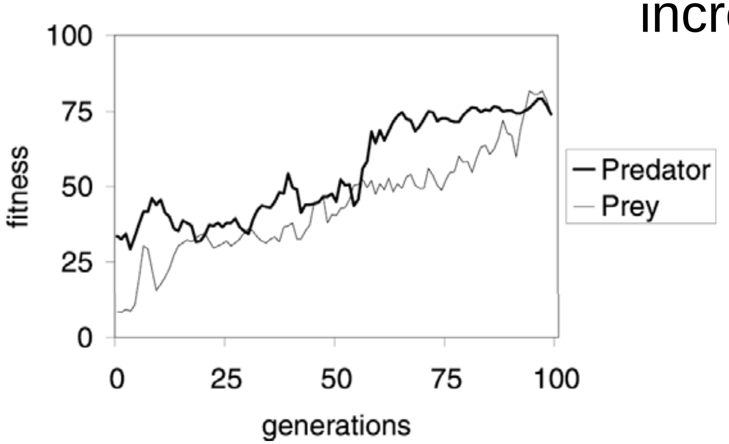
When we measure fitness of evolved predator robots using distance, we see that they do not attemp to optimize it. Our results indicate that co-evolution may work better with internal, implicit, and behavioral fitness functions.





Hall of Fame



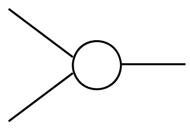


In order to avoid the cycling dynamics of competitive co-evolution, Rosin and Belew [1997] suggested to store all best individuals (*Hall of Fame*) and test each new individual against all best opponents obtained so far. Using this methods, the number of tournaments increases along generations.

It turns out that it is sufficient to test new individuals only against a limited sample (10, e.g.) randomly extracted from the Hall of Fame in order ot produce continuous incremental progress, as shown by CIAO and Master graphs.

In the long run, Hall of Fame becomes equal to single-agent evolution because the pool of opponents does not change. In other words, the potential for creative new solutions becomes smaller.

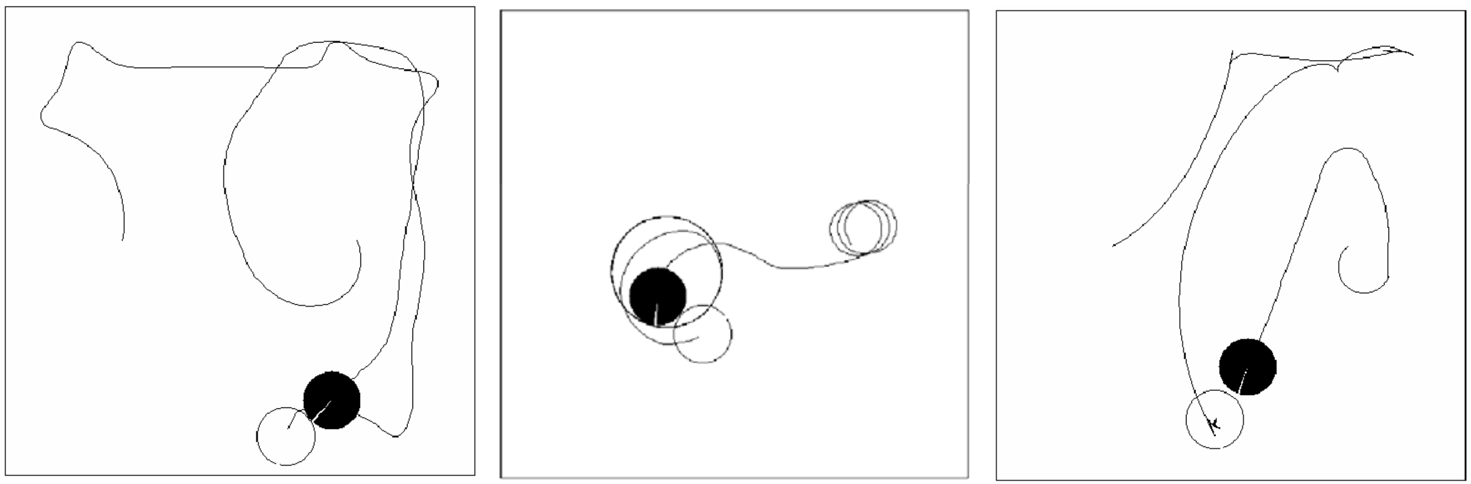
Allowing life-long adaptation



GENES ENCODE:
- weights
- hebb rules
- random values

Co-evolutionary dynamics are drastically changed:

- After 20 generations, predators always win
- Predators always choose adaptation (hebb rules)
- Prey most often choose random synapses
- Adaptation does not help prey because of poor sensors

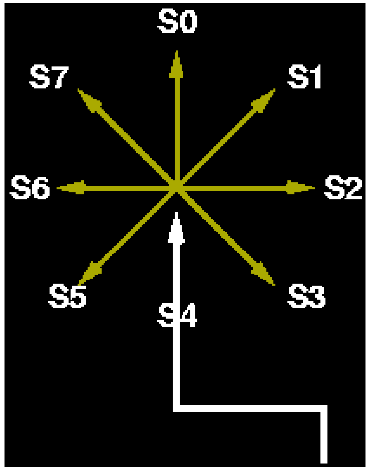


Man-machine co-evolution

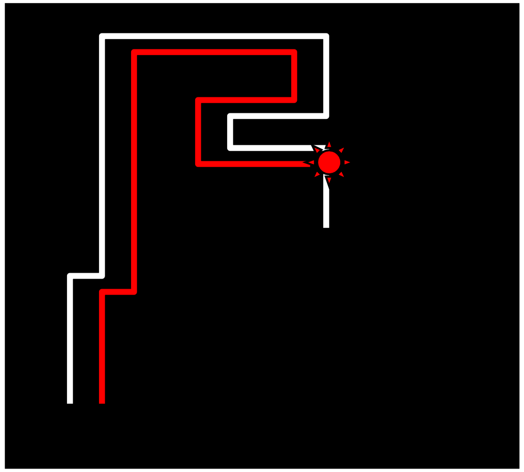
Funes & Pollack [2000] co-evolved computer programs and human players on a simplified version of the game Tron.

Computer programs are represented as trees and evolved using GP. Those programs that win against human players have higher probability of reproducing. Instead, humans are free to decide whether to play or not.

Tron, 1982, Walt Disney Pictures



Computer Agent
Sensory Information

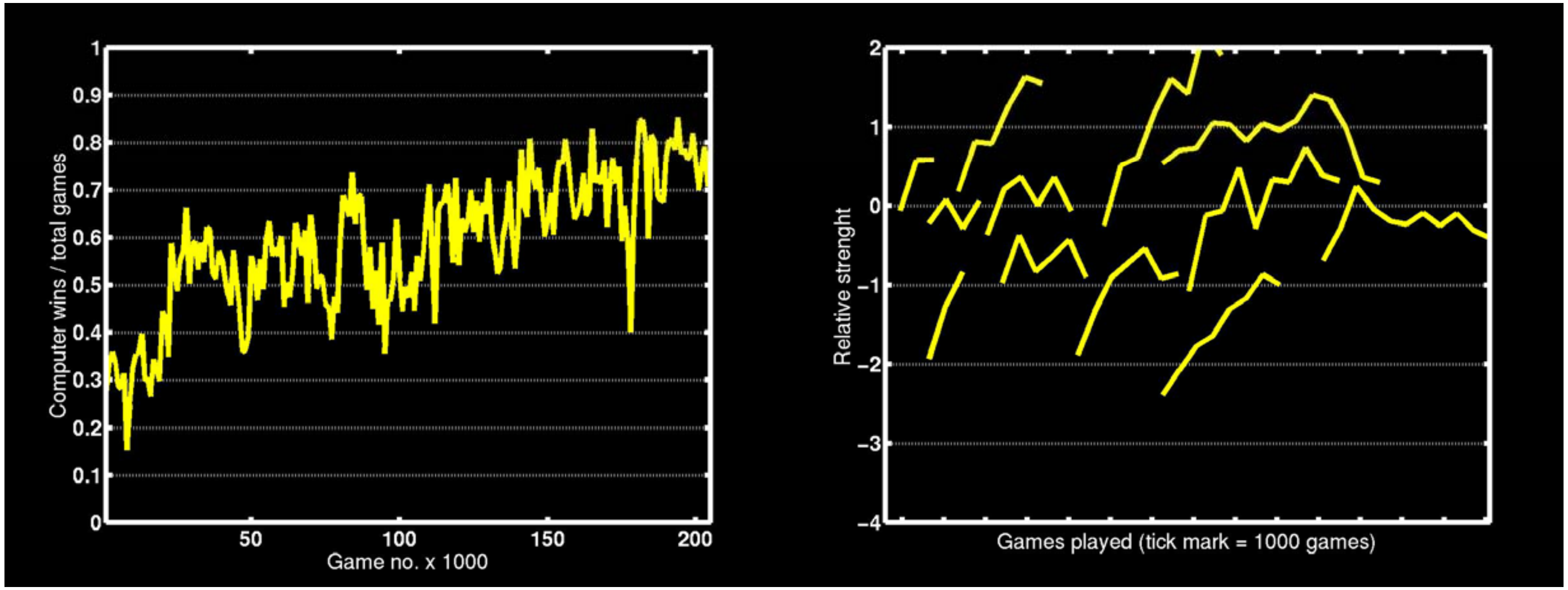


Game Snapshot

Machines win

The major results are that:

- Computer programs become increasingly better and hard to defeat
- Human population does not evolve
- Human individuals learn across trials

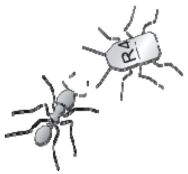


Evolution of cooperation

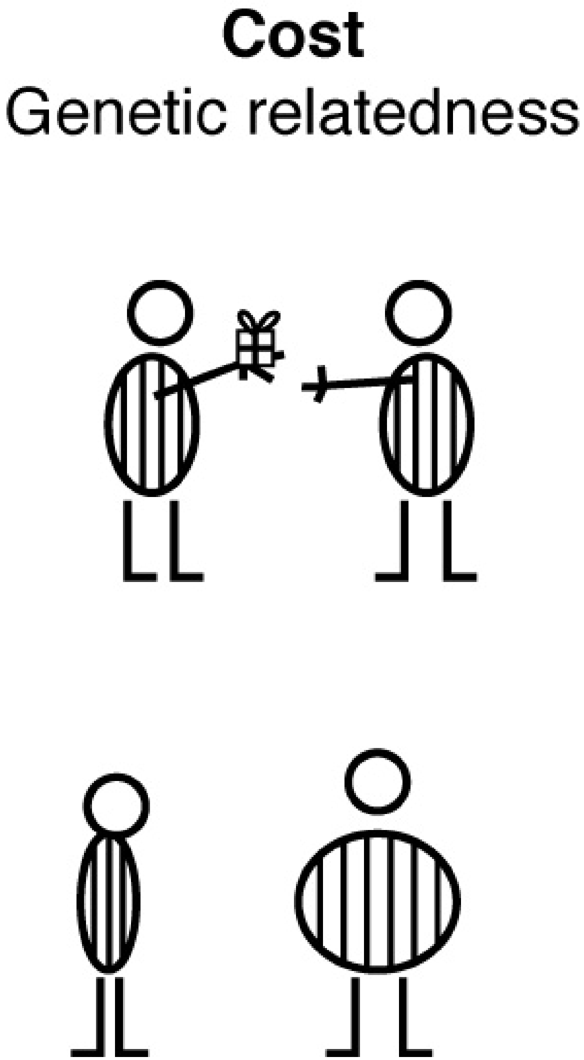
Cooperation can easily evolve if there is an advantage and no cost in helping somebody else because the fitness of individuals is increased

Photo: Pete Ellison

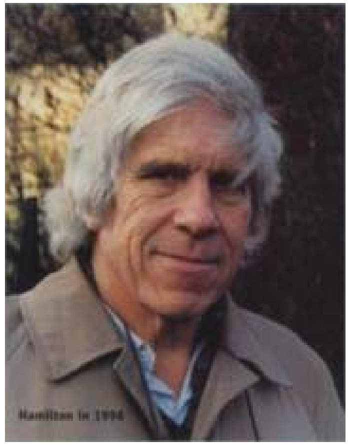
Altruistic cooperation is difficult to explain because it involves a cost for the individual. Example: Warrior ants that die to save the colony

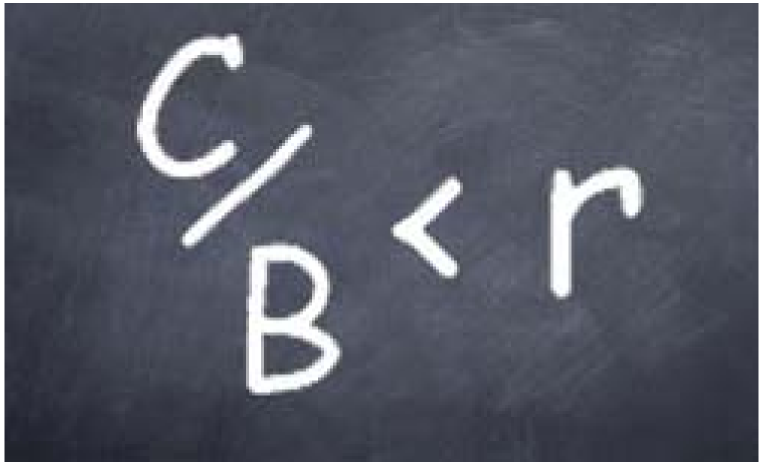


Genetic relatedness

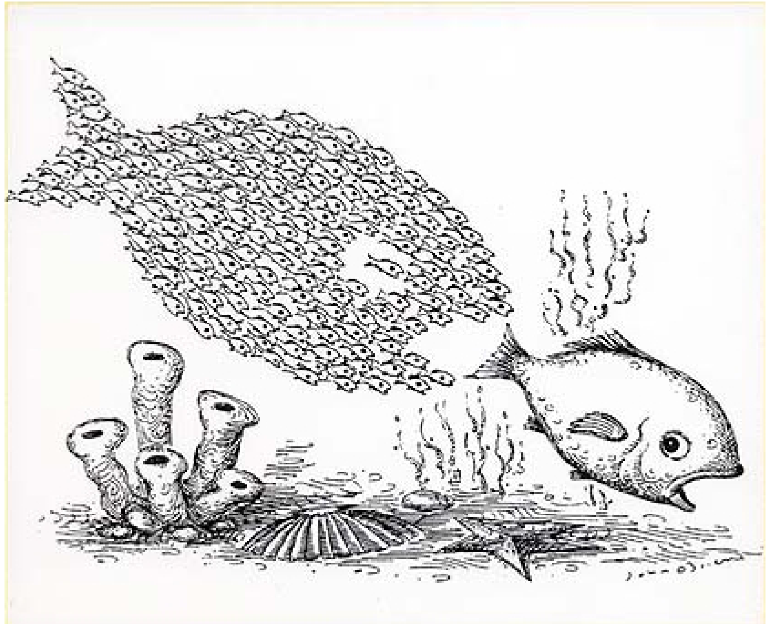


Hamilton (1964)





Group selection



E.g.: Wynne-Edwards (1986);
Michod (1999)

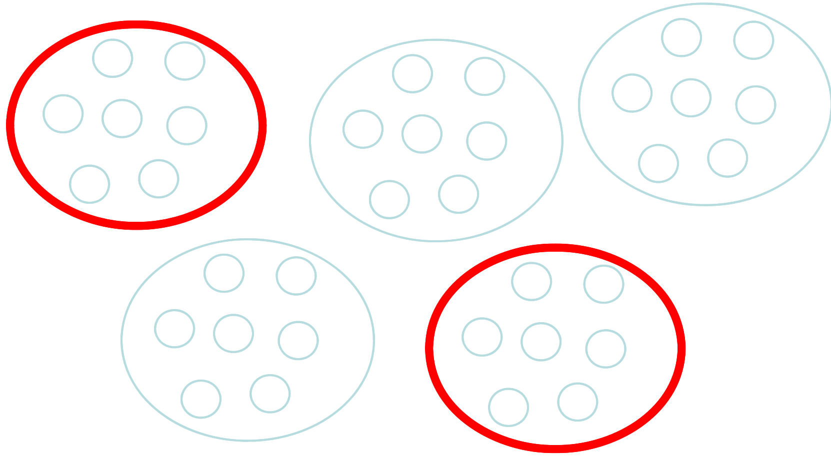
No need for genetic relatedness (but Wolpert & Szathmary, 2002)
Criticism: Mutation at the level of the group slower / less likely
Recent findings: Mutation silencing

In Artificial Evolution, no need to compute individual fitness

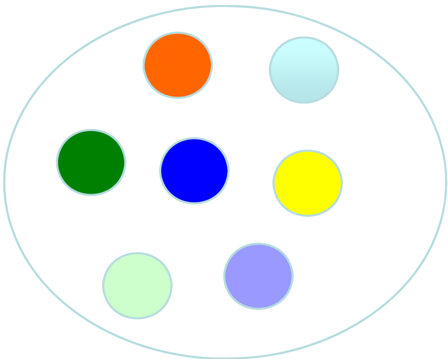
Four possible algorithms

HOMOGENEOUS

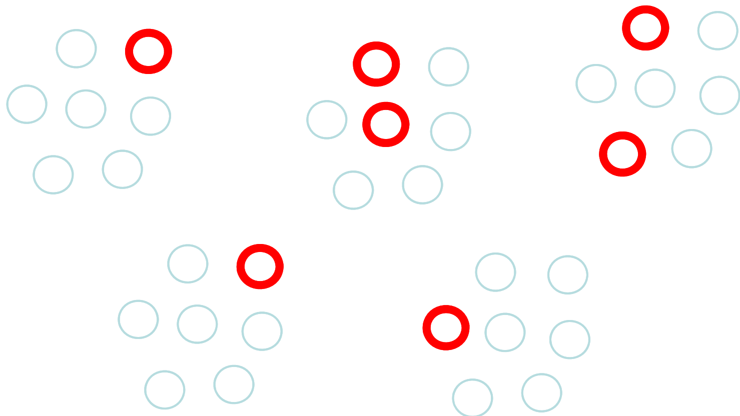
TEAM SELECTION



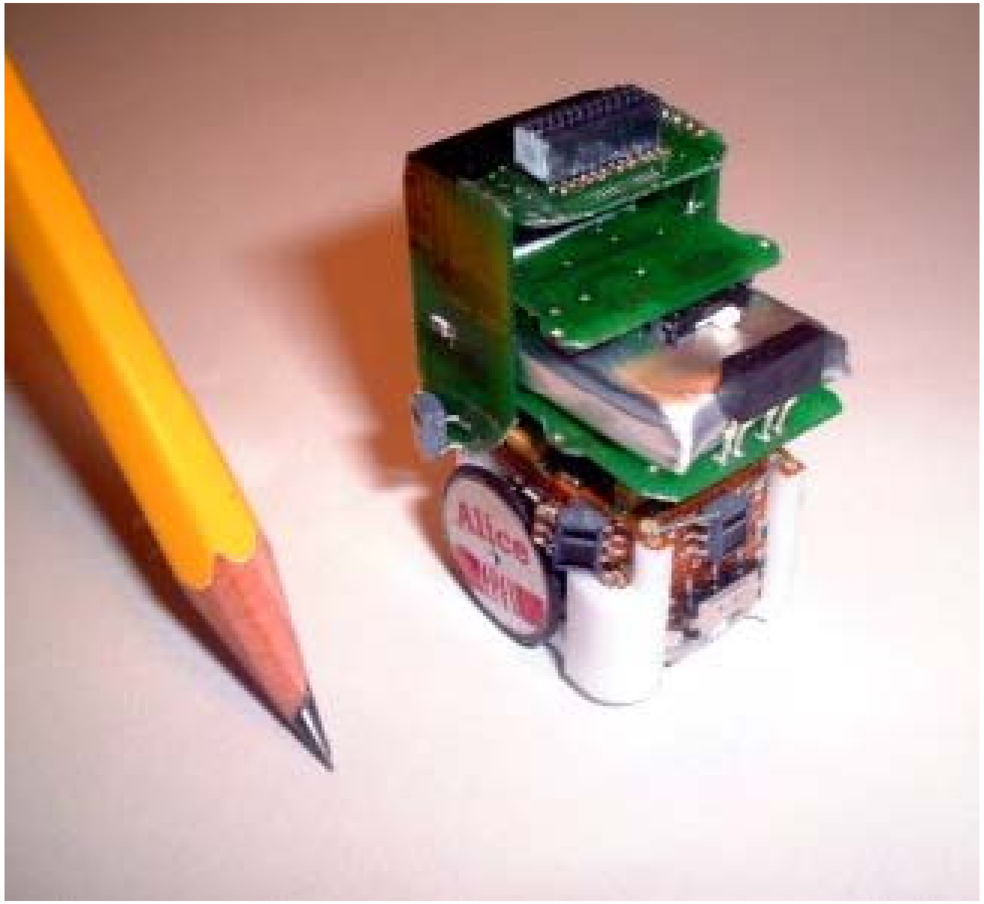
HETEROGENEOUS



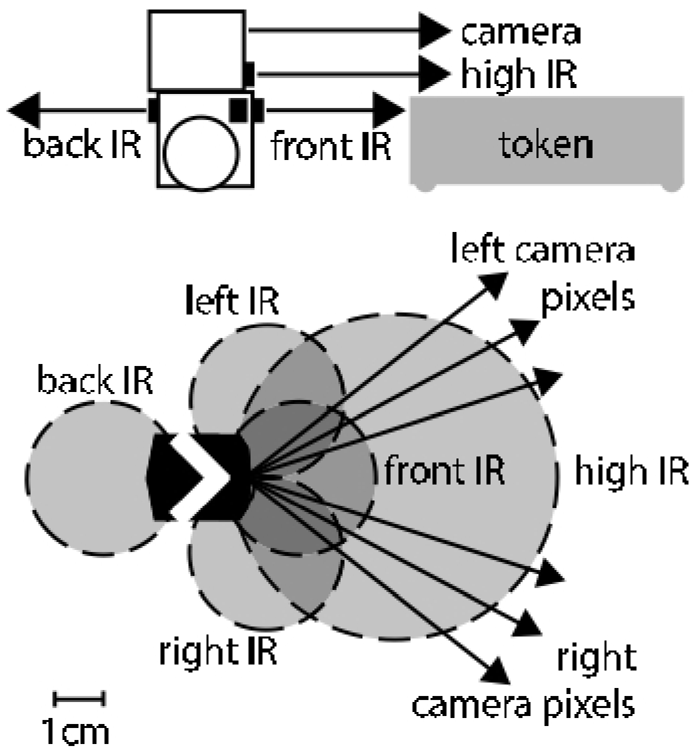
INDIVIDUAL SELECTION

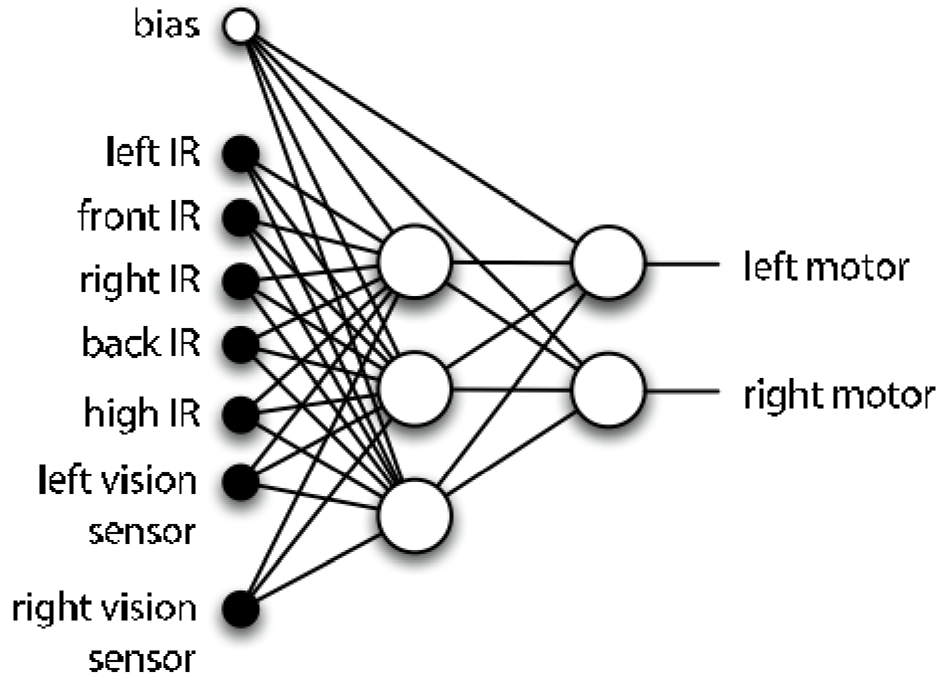


Robot Foraging Task



Control structure

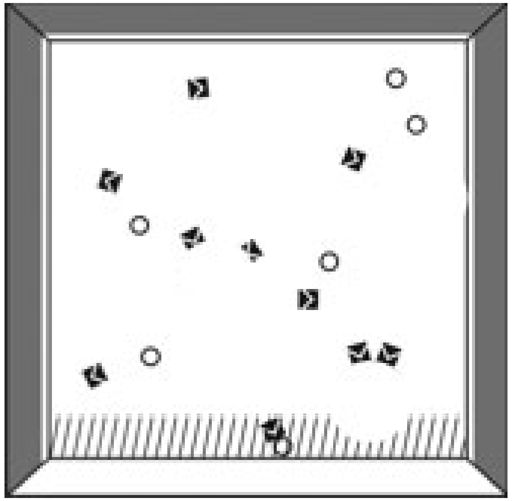




Connection weights of network are encoded in artificial genome
Each team is composed of 10 robots
The population is composed of 100 teams
Each team is evaluated 10 times

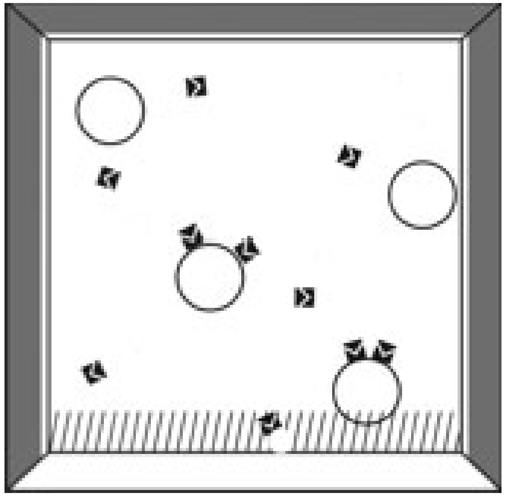
Types of tasks

INDIVIDUAL



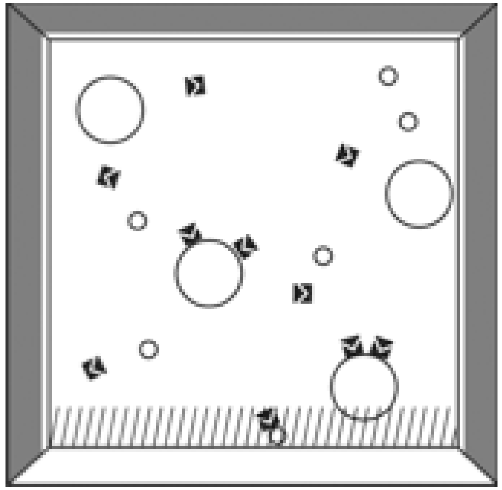
1 fitness point per object to foraging robot

COOPERATIVE



1 fitness point to **all** robots for each object (2 robots necessary to push an object)

ALTRUISTIC

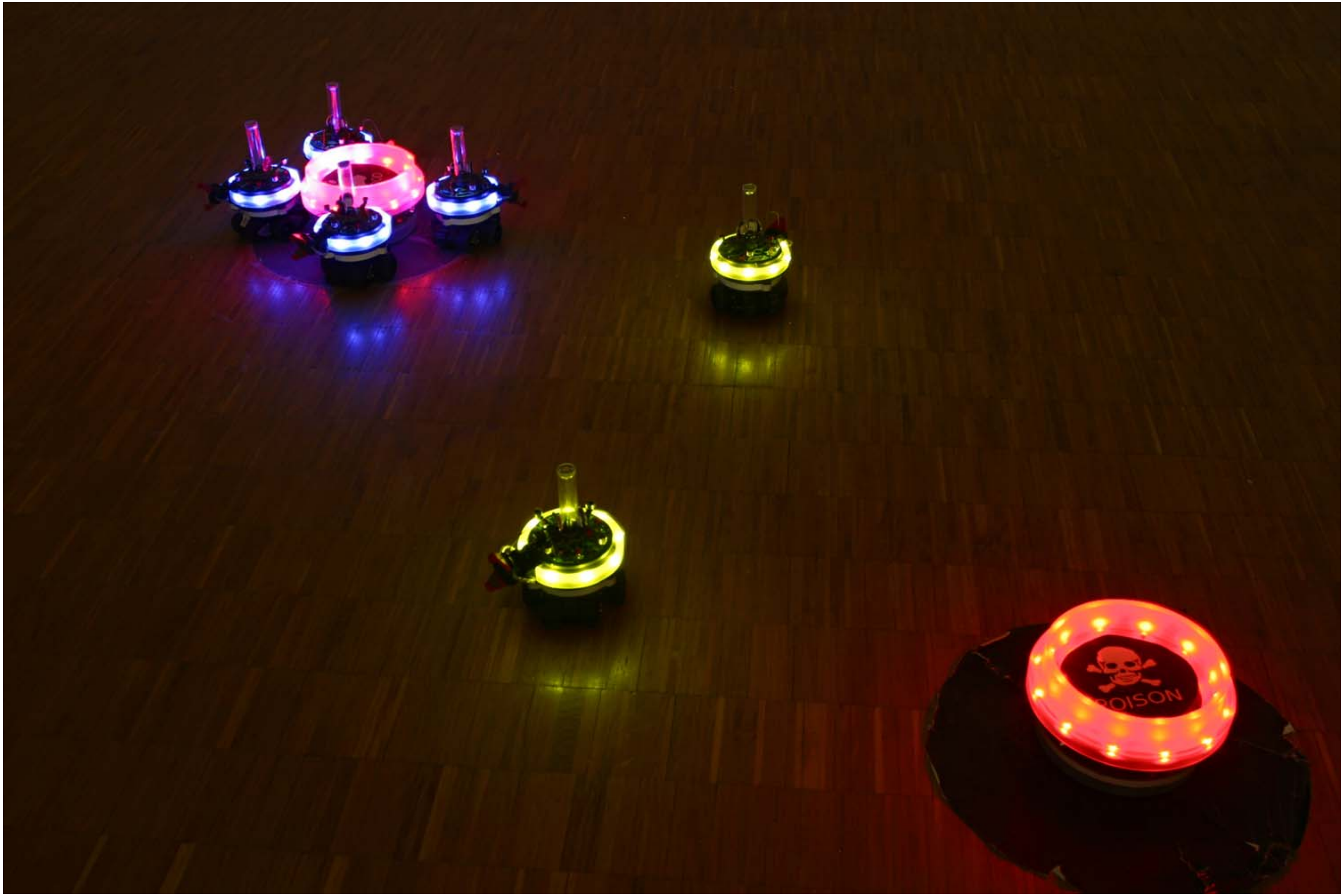


1 fitness point to **all** robots for each large object

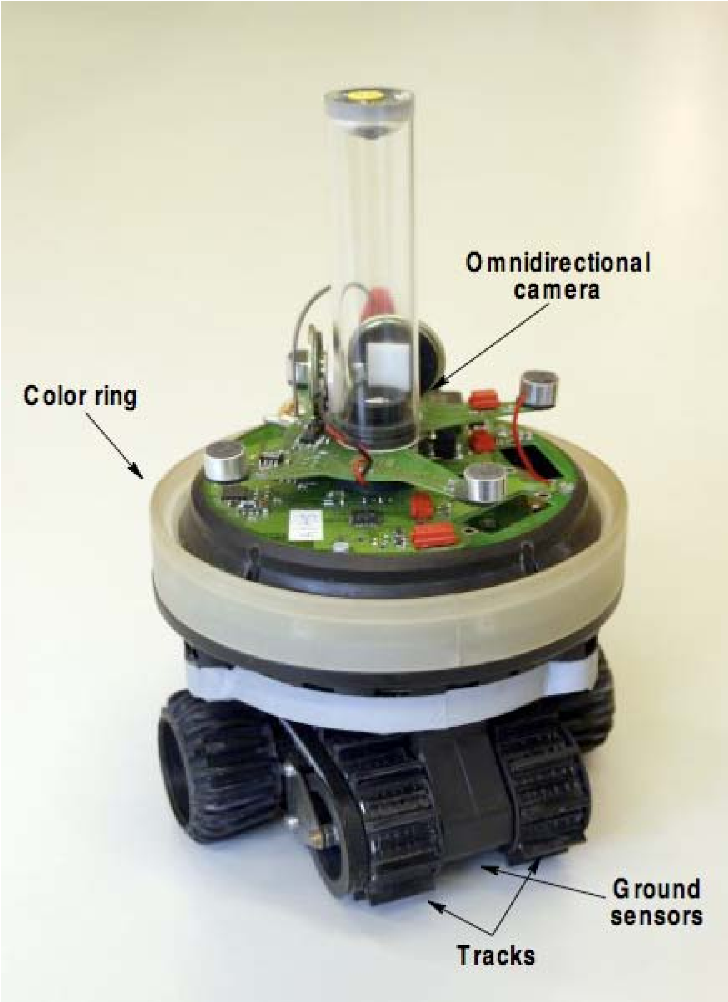
1 fitness point per small object to individual robot

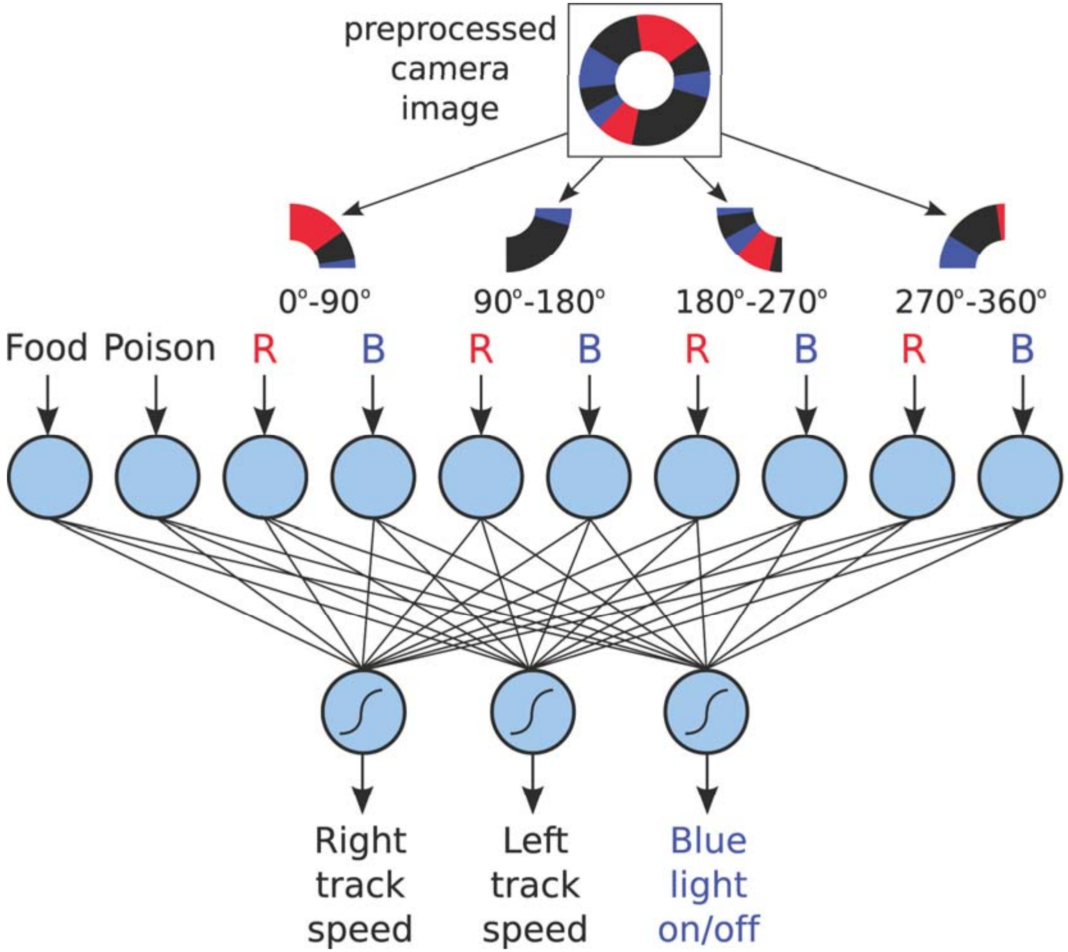
Genetically related, group selection

Foraging with Uncertainty

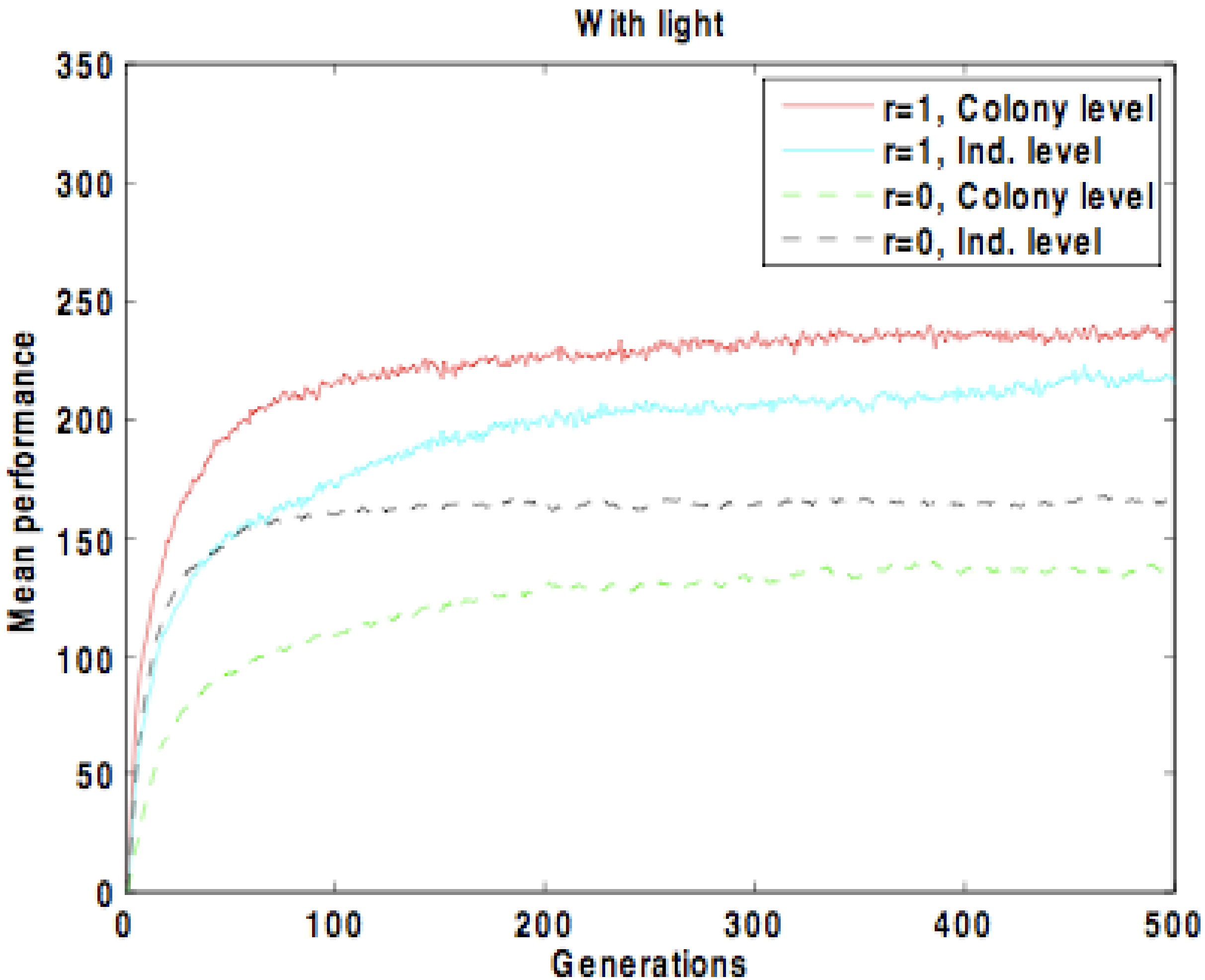


Control structure



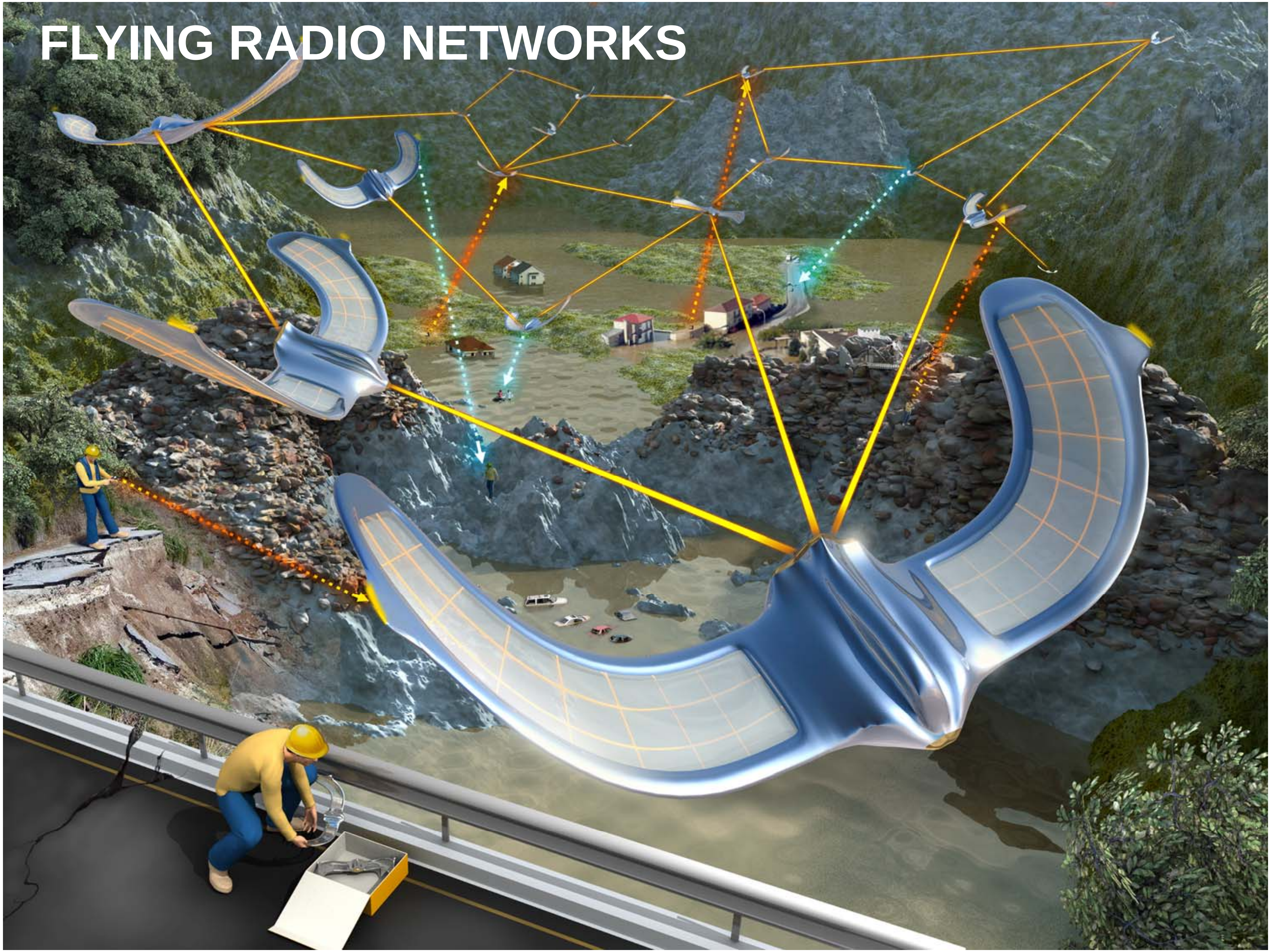


Comparative Performance

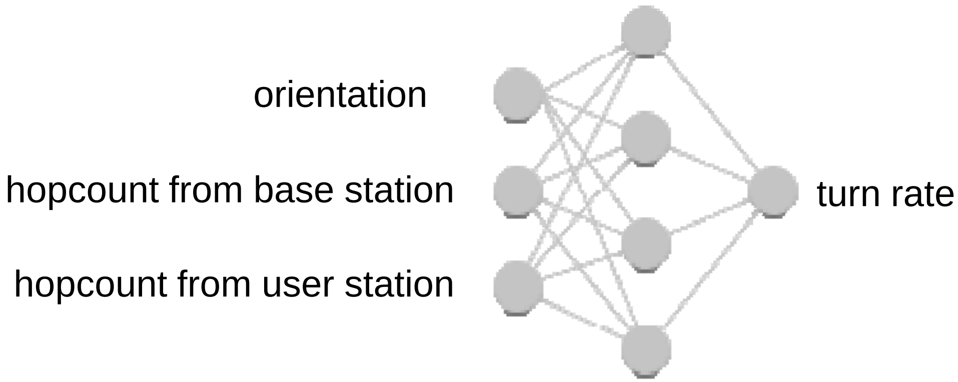


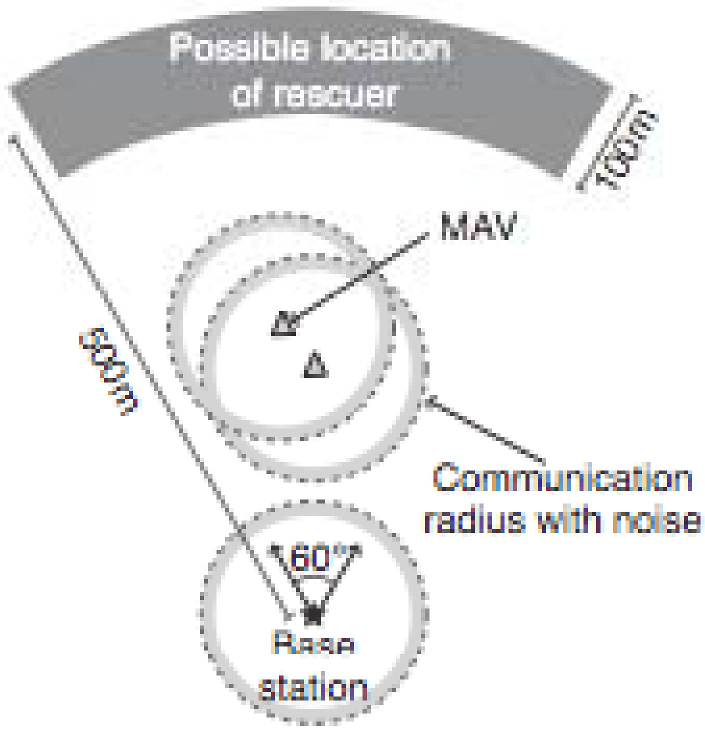
Genetically related individuals obtain highest performance

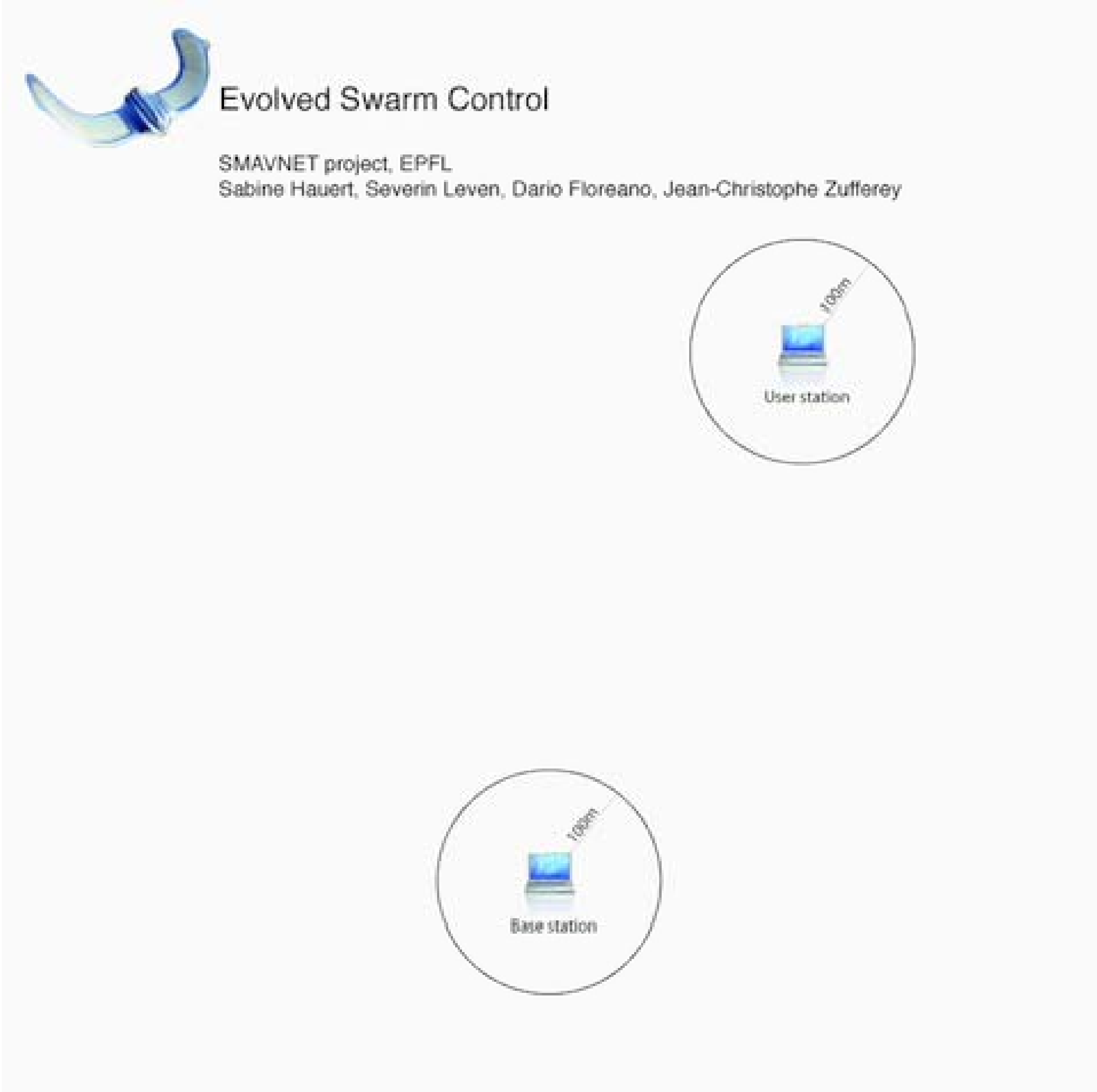
EVOLVED IN HARDWARE


FLYING RADIO NETWORKS

Evolutionary Conditions






Evolved Swarm Control
SMAVNET project, EPFL
Sabine Hauert, Severin Leven, Dario Floreano, Jean-Christophe Zufferey
100m
User station
100m
Base station

Summary

Competitive co-evolution can potentially create more efficient and novel systems

It is hard to harness and direct it towards desired solutions (extrinsic fitnesses limit co-evolutionary dynamics)

Generational memory is useful for preventing or retarding recycling

Altruistic cooperation evolves if individuals are genetically related or there is group-level selection