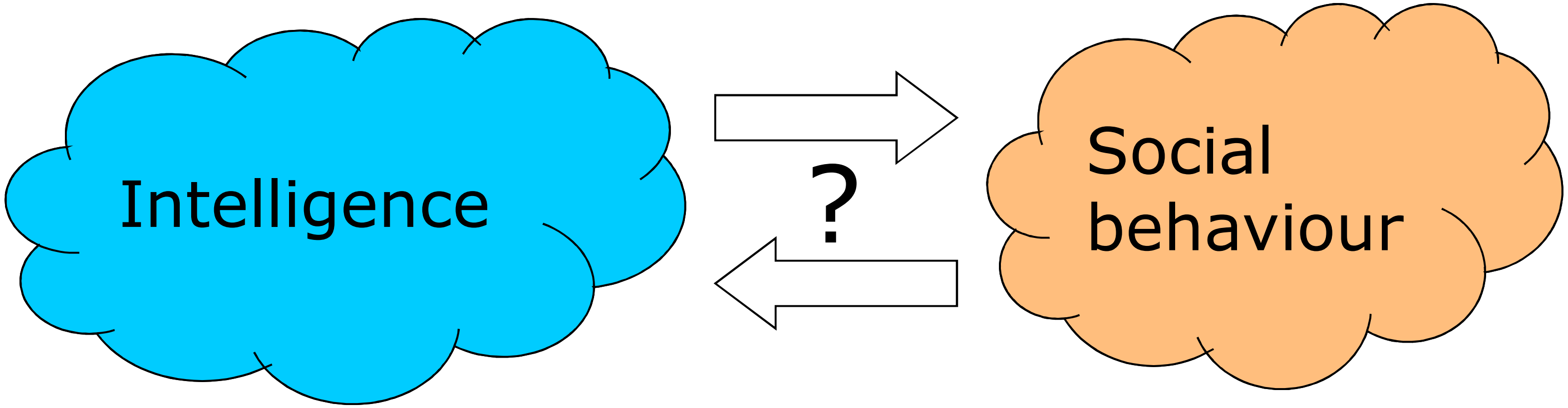
Swarm Intelligence

Bacteria, ants, birds and fish in computing

Olle Gällmo

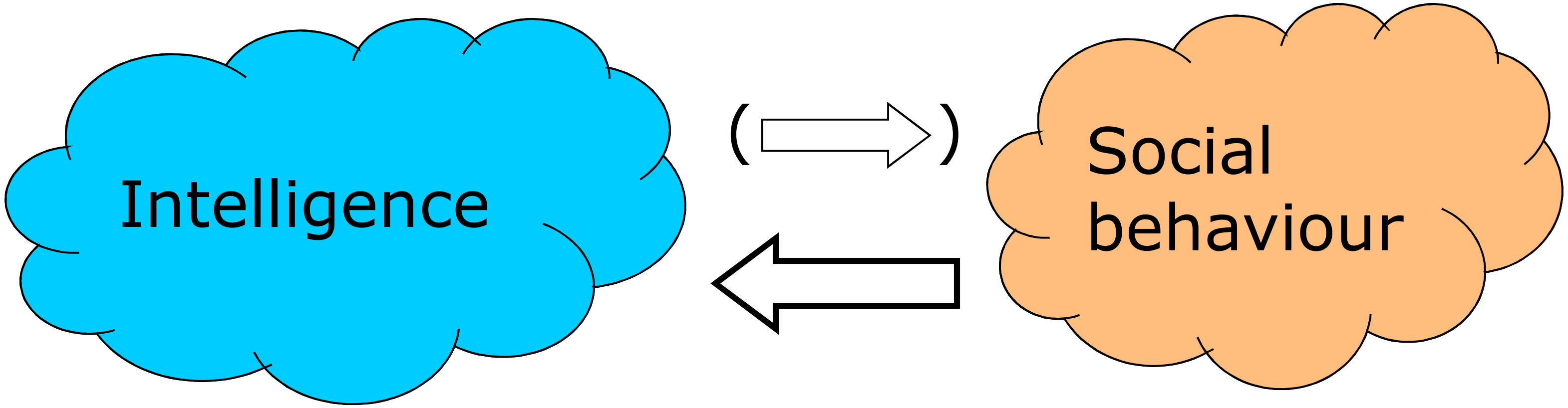
Swarm Intelligence



- Intelligence (artificial and natural) is often considered a property of *individuals*
- Most intelligent animal species are social
- Are we social because we are intelligent or is it the other way around?
  - Both, of course (they are co-evolving), but one direction has been studied longer than the other

Olle Gällmo | olle.gallmo@it.uu.se

Swarm Intelligence



- Intelligence can emerge from social interaction
- *Emergent behaviour* – when a group behaves in ways that were not ”programmed” into its members
- Swarm intelligence
  - simulated social interaction
  - emergent collective intelligence in groups of simple agents

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

Observations

- Bird flocks and fish schools move in a coordinated way, but there is no coordinator (leader)
  - So, what decides the behaviour of a leader-less flock?
- Ants and termites quickly find a short path between the nest and a food source
  - ... and solve many other advanced problems as well
    - keeping cattle, building (ventilated) housing, coordinated heavy transports, tactical warfare, cleaning house, etc.
  - A single ant is essentially a blind, memory-less, random walker!
- Distributed systems without central control
- Useful not only to simulate but also to solve optimization problems

Olle Gällmo | olle.gallmo@it.uu.se

Bird flocks and fish schools

- Local interaction
- No leader
- Simple local rules – a weighted combination of several goals
  - match velocity of neighbours
  - avoid collisions with neighbours
  - avoid getting too far from neighbours
    - or strive for centre of the flock (fish)
- Sufficient to make very realistic simulations
  - used in movies and computer graphics
  - remove the match-velocity rule: insect swarm
  - remove collision rule: cultural interaction

Olle Gällmo | olle.gallmo@it.uu.se

Stampede in ”Lion King”

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

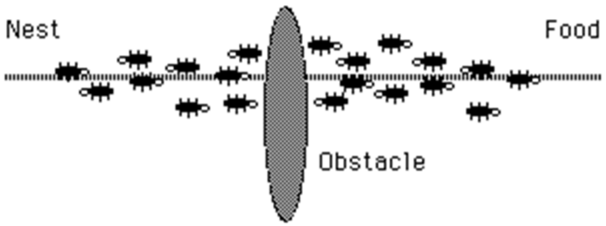
What about the ants?

- How do they search for the shortest route?
  - Individual ants don't
  - The colony does
- Ant colonies are much more intelligent than ants
  - Ant colonies adapt, ants don't (much)
  - Ants have almost no memory and can not build cognitive maps. Ant colonies can (and do)
    - Mammals build cognitive maps in their brains
    - Ant colonies build them in their environment, through pheromone trails
- Ants are better thought of as cells in a greater organism – the colony
  - Also without leader – the queen is not a controller

Olle Gällmo | olle.gallmo@it.uu.se

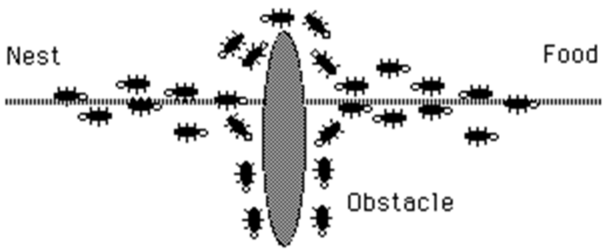
Ants find shortest paths

Ant path

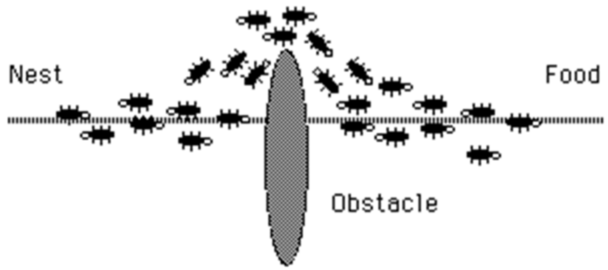


An obstacle appears

At first, the ants select at random



After a while, pheromones become more concentrated on the shortest route



Drawings by Marco Dorigo

Olle Gällmo | olle.gallmo@it.uu.se

Stigmergy

Indirect communication and coordination, by local modification and sensing of the environment

Olle Gällmo | olle.gallmo@it.uu.se

Walls, tunnels and bridges

UPPSALA UNIVERSITET

Computational tools

- Cellular automata
  - 1940's (von Neumann et al)
  - an alternative computer architecture
- Ant Colony Optimization
  - 1991 (Dorigo)
  - mostly for combinatorial optimization
- Particle Swarm Optimization
  - 1995 (Kennedy & Eberhart)
  - more general optimization technique

Olle Gällmo | olle.gallmo@it.uu.se

Computational tools

- Cellular automata
  - 1940's (von Neumann et al)
  - an alternative computer architecture
- Ant Colony Optimization
  - 1991 (Dorigo)
  - mostly for combinatorial optimization
- Particle Swarm Optimization
  - 1995 (Kennedy & Eberhart)
  - more general optimization technique

Olle Gällmo | olle.gallmo@it.uu.se

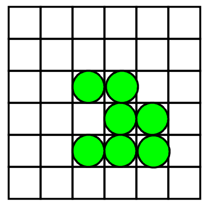
Cellular automata

- Massively parallel system of identical communicating state machines (cells)
- A cell's *state* (e.g. on/off) is a function of the states of the cells it communicates with (its neighbours)
  - The neighbourhood is usually topolocial
- Used to model/animate fluids (e.g. the water in *Find Nemo*), gases, bacterial growth, swaying grass, social interaction, epidemics, in ecological simulations etc.

Olle Gällmo | olle.gallmo@it.uu.se

Conway's Game of Life

- World: a 2D grid. Each square represents a cell
- States: Living or dead
- Neighbourhood: The eight surrounding cells
- Initialize with a random number of living cells
- State transition rules:
  - A living cell with <2 living neighbours dies (loneliness)
  - A living cell with >3 living neighbours dies (overcrowded)
  - A dead cell with exactly 3 living neighbours comes alive
  - All other cells keep their current state



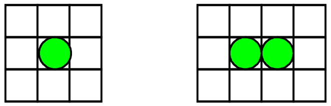
Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

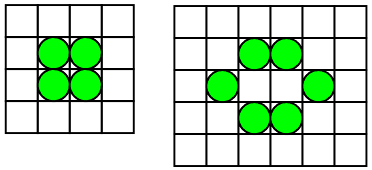
Conway's Game of Life

- State transition rules:
  - A living cell with <2 living neighbours dies (loneliness)
  - A living cell with >3 living neighbours dies (overcrowded)
  - A dead cell with exactly 3 living neighbours comes alive
  - All other cells keep their current state

Will die



Stationary

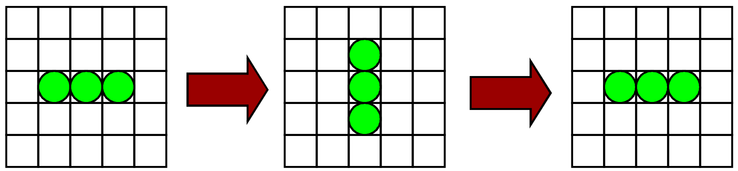


Olle Gällmo | olle.gallmo@it.uu.se

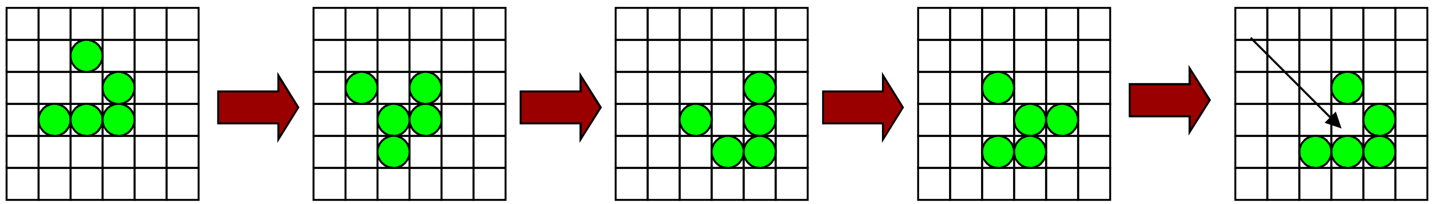
Conway's Game of Life

- State transition rules:
  - A living cell with <2 living neighbours dies (loneliness)
  - A living cell with >3 living neighbours dies (overcrowded)
  - A dead cell with exactly 3 living neighbours comes alive
  - All other cells keep their current state

Cyclic:

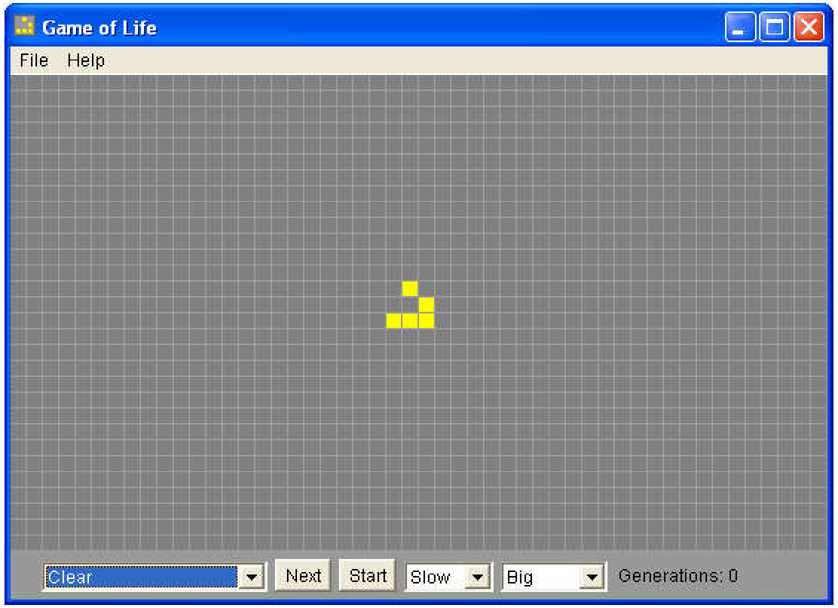


Cyclic and mobile



Olle Gällmo | olle.gallmo@it.uu.se

Life demo



Game of Life
File Help
Clear
Next
Start
Slow
Big
Generations: 0

UPPSALA UNIVERSITET

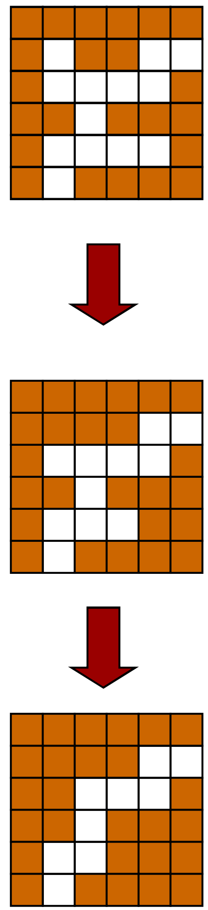
Observations

- Simple (and deterministic!) rules – complex emergent behaviour
- Few rule sets have this property
  - Why?
- GA/GP to find new rules?
- Fixed set of rules (e.g. Conways)
  - still universal
  - depends on initial cell configuration
- One-way function (cryptography)
- Natural Computation, but is it ML?

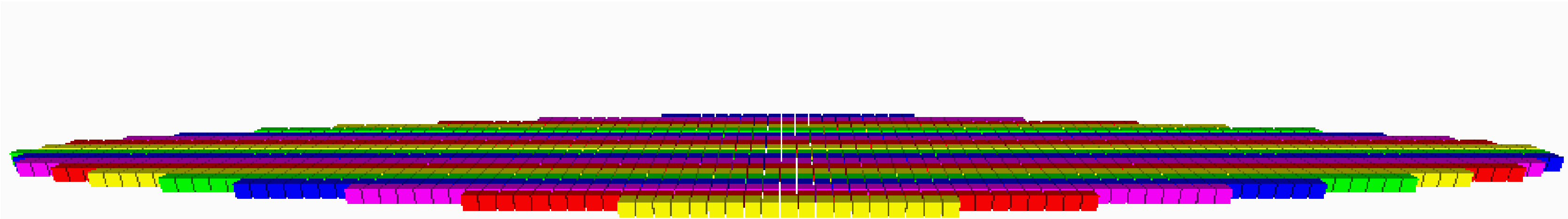
Olle Gällmo | olle.gallmo@it.uu.se

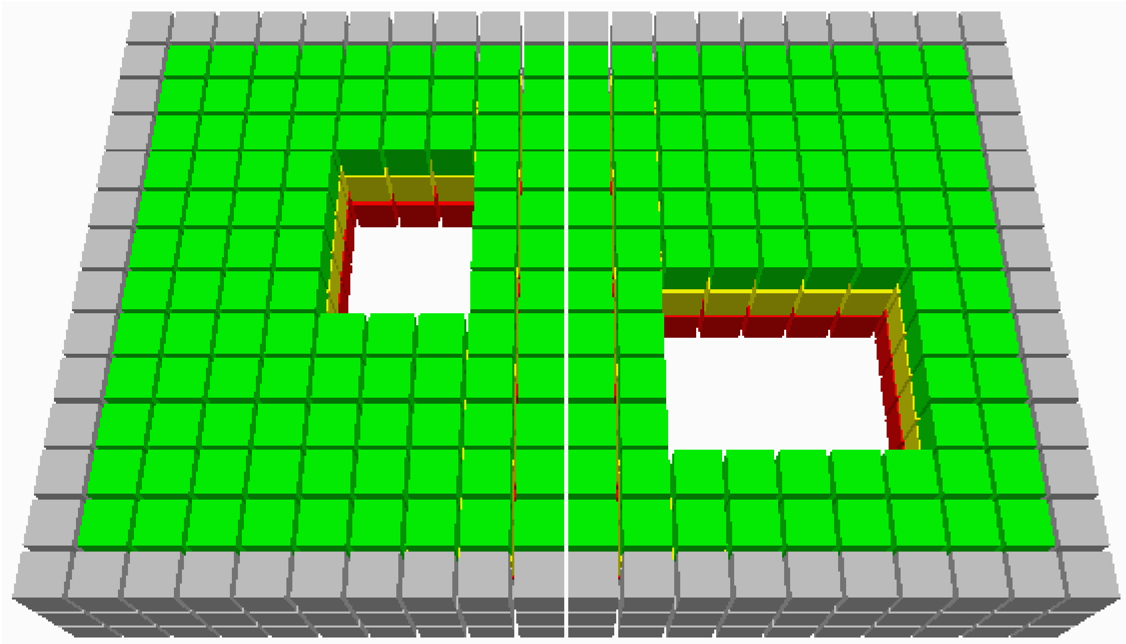
CA and maze problems

- World: 2D grid which overlays the maze
- States: Corridor or wall
- Neighbourhood: The four surrounding cells (n, e, s, w)
- Initialize cells according to the maze
- State transition rules:
  - A corridor cell with 3 or 4 neighbouring wall cells becomes a wall cell
  - A wall cell remains a wall cell
- Terminates after *n* generations, where *n* is the length of the longest blind alley
- Only cells on a route between start and goal remain corridor cells
- Can extend to include several goals, finding the shortest route, etc.



Construction and repair

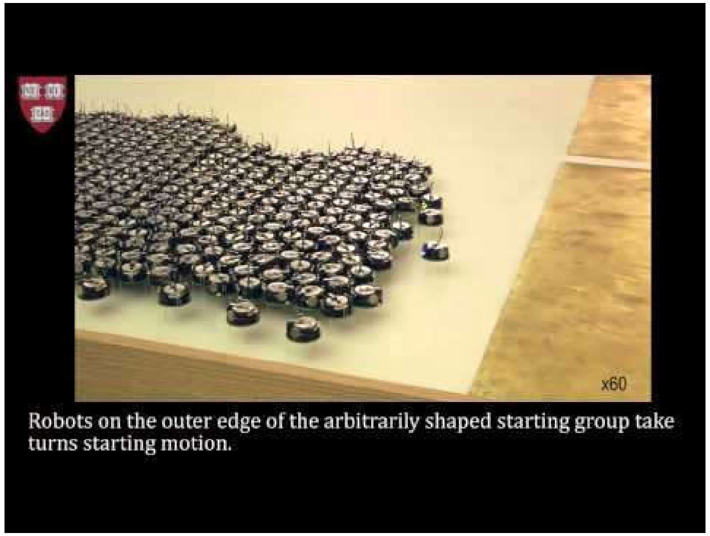




Olle Gällmo | olle.gallmo@it.uu.se

Other applications

- Collaborating simple robots
  - Locomotion
  - Space probes
- Modelling
  - Water, avalanches, traffic flows, ...
- Map/level generators for games



Olle Gällmo | olle.gallmo@it.uu.se

Computational tools

- Cellular automata
  - 1940's (von Neumann et al)
  - an alternative computer architecture
- Ant Colony Optimization
  - 1991 (Dorigo)
  - mostly for combinatorial optimization
- Particle Swarm Optimization
  - 1995 (Kennedy & Eberhart)
  - more general optimization technique

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

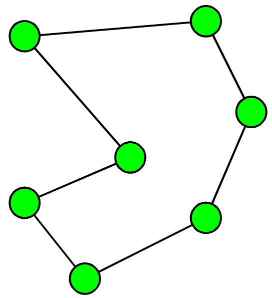
Ant Colony Optimization

- Family of combinatorial optimization algorithms, based on ant behaviour
- Common benchmark: the Travelling Salesman Problem (TSP)
- Common ’real’ applications
  - Scheduling and
  - Network routing (AntNet)
- Members: ACS, Ant-Q, MMAS, AS_{rank}, ...
  - most of which are extensions to Dorigo’s Ant System (AS)

Olle Gällmo | olle.gallmo@it.uu.se

Traveling Salesman Problems (TSP)

- Find the shourtest tour through N cities, and then back to the starting point, such that
  - each city is visited once and only once
- NP-hard
  - (N-1)!/2 possible tours
  - exhaustive search intractable
- Specialized algorithms exist
  - and are hard to beat



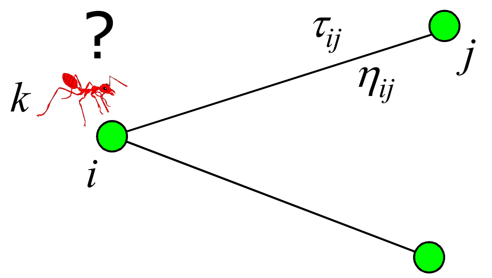
Olle Gällmo | olle.gallmo@it.uu.se

Ant System for TSP

Each ant (k)

- is placed in a randomly selected city
- remembers the partial solution found so far (initially, the start city only)
- moves stochastically from city (i) to city (j), by some transition probability

$$p_{ij}^{k}(t)$$



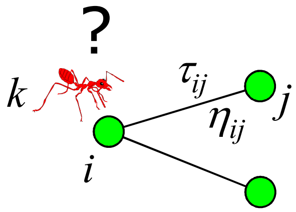
which depends on

- pheromone intensity, τ_{ij}
- local information, η_{ij} (distance)
- whether j is feasible (not already visited)

Olle Gällmo | olle.gallmo@it.uu.se

Transition probabilities

Pheromone intensity, at time t, on the path from city i to j



Local information: In TSP $\eta_{ij}=1/d_{ij}$, where d_{ij} is the distance between city i and j

$$p_{ij}^{k}(t)=\frac{\left[\tau_{ij}(t)\right]^{\alpha}*\left[\eta_{ij}\right]^{\beta}}{\sum_{c\in C_i^k}\left[\tau_{ic}(t)\right]^{\alpha}*\left[\eta_{ic}\right]^{\beta}},\ j\in C_i^k$$

Probability, at time t, of ant k traveling from city i to city j

Set of feasible destination cities (directly reachable from city i, and not yet visited by ant k)

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

Effects of α and β

$$p_{ij}^{k}(t)=\frac{\left[\tau_{ij}(t)\right]^{\alpha} * \left[\eta_{ij}\right]^{\beta}}{\sum_{c \in C_i^k}\left[\tau_{ic}(t)\right]^{\alpha} * \left[\eta_{ic}\right]^{\beta}}, j \in C_i^k$$

- If α=0, β>0
  - Pheromone information discarded, only local info used
  - Stochastic greedy search with multiple starting points
- If α>0, β=0
  - No local information used, only pheromones
    - more like real ants (?)
  - May lead to premature convergence
    - all ants tend to follow the same (suboptimal) route
    - difficult to discover new shortcuts (as for real ants)

Olle Gällmo | olle.gallmo@it.uu.se

Pheromone update

When all ants have completed a tour, let each ant deposit pheromones on the paths it followed

Evaporation rate $(0<\rho\leq 1)$

Sum over all ants, k

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_k \Delta\tau_{ij}^k(t) \quad \forall (i,j)$$

ant k's contribution

$$\Delta\tau_{ij}^k(t) = \begin{cases} 1/L_k(t) & \text{if path } ij \text{ was used by ant } k \\ 0 & \text{otherwise} \end{cases}$$

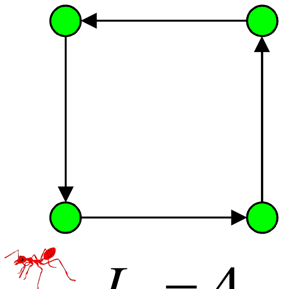
$L_k(t)$ = length of ant k's tour

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

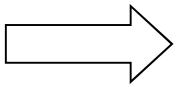
Trivial example (4 cities)

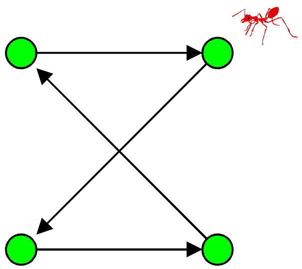
Two ants



$$L_1 = 4$$

$$\Delta\tau_{ij}^k(t) = 1/L_k(t) = 1/4 = 0.25$$

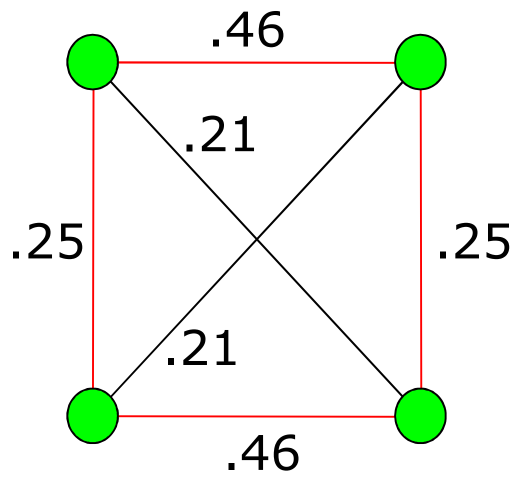




$$L_2 = 2 + 2 * \sqrt{2} \approx 4.8$$

$$\Delta\tau_{ij}^k(t) = 1/L_k(t) = 1/4.8 \approx 0.21$$

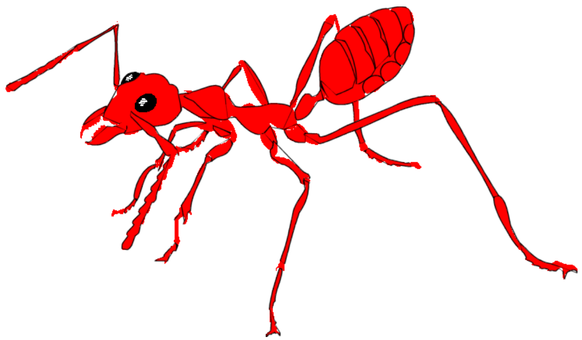
τ



Olle Gällmo | olle.gallmo@it.uu.se

Ant System TSP Demo

- 20 cities (19!/2 = $6.1*10^{16}$ possible tours)
- 20 ants (one in each city)
- $\alpha=\beta=1$
- Evaporation rate, $\rho=0.9$



Olle Gällmo | olle.gallmo@it.uu.se

Notes on Ant Colony Opt.

- Not really a swarm?
  - These ants are not aware of each other, only of pheromones and other local info
- No direct communication $\Rightarrow$ very scalable!
- The TSP solution demonstrated here works, but is not state-of-the-art
  - Best ACO algorithms exploit available global information
- ACO is most promising for non-stationary problems (e.g. network routing)
  - fewer competitors

Olle Gällmo | olle.gallmo@it.uu.se

What is "optimal"?

- Specialized algorithms v.s. general "black-box" ones
- Problem oriented def. of "optimal"
  - Specialized algorithms usually wins
- In practice (in industry), 'optimality' involves other concerns as well
  - time and cost to setup and maintain
  - amount of knowledge required
  - good enough is good enough

Olle Gällmo | olle.gallmo@it.uu.se

Pragmatic advice

Your algorithms teacher would probably not agree, but

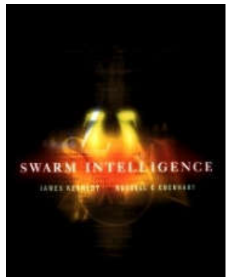
It is often better to use an algorithm/method you know well, than to search for (and tune) the "best" one!

But, of course, if you happen to know the best one ...

Olle Gällmo | olle.gallmo@it.uu.se

Computational tools

- Cellular automata
  - 1940’s (von Neumann et al)
  - an alternative computer architecture
- Ant Colony Optimization
  - 1991 (Dorigo)
  - mostly for combinatorial optimization
- Particle Swarm Optimization
  - Kennedy & Eberhart
    - *Swarm Intelligence*, Morgan Kaufmann, 1995



  - a more general optimization technique

UPPSALA UNIVERSITET

Particle Swarm Optimization

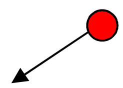
- Originally intended to simulate bird flocks and to model social interaction
  - but stands on its own as an optimization tool
- A population of *particles*
  - Population sizes, typically 10-50 (smaller than in EC)
- A particle, i, has a position, $\boldsymbol{x}_i$, and a velocity, $\boldsymbol{v}_i$
  - Both vectors in n-dimensional space
- Each particle's position, $\boldsymbol{x}_i$, represents one solution to the problem
- Each particle remembers the best position it has found, so far, $\boldsymbol{p}_i$

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA
UNIVERSITET

The flying particle

- The particles ”fly” through n-dimensional space, in search for the best solution



$$x_{i,d}(t) = x_{i,d}(t\text{-}1) + v_{i,d}(t)$$

- The velocities, $\boldsymbol{v}$, depend on previous experience of this particle and that of its neighbours

  Discrete time $\Rightarrow$ velocity = step length

- Neighbourhood definition varies
  - Extreme cases: pbest (personal) and gbest (global)
  - General case: lbest (local best)
    - pbest and gbest are special cases

Olle Gällmo | olle.gallmo@it.uu.se

Personal best (pbest)

- No interaction between particles

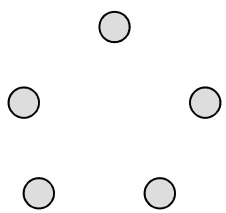
For all particles, i, and all dimensions, d:

$$v_{i,d}(t) = v_{i,d}(t-1) + \mathrm{U}(0,\varphi) * (p_{i,d} - x_{i,d}(t-1))$$

Uniformly distributed random number

acceleration constant (typically ≈ 2)

best position found so far by particle *i*



Neighbourhood structure

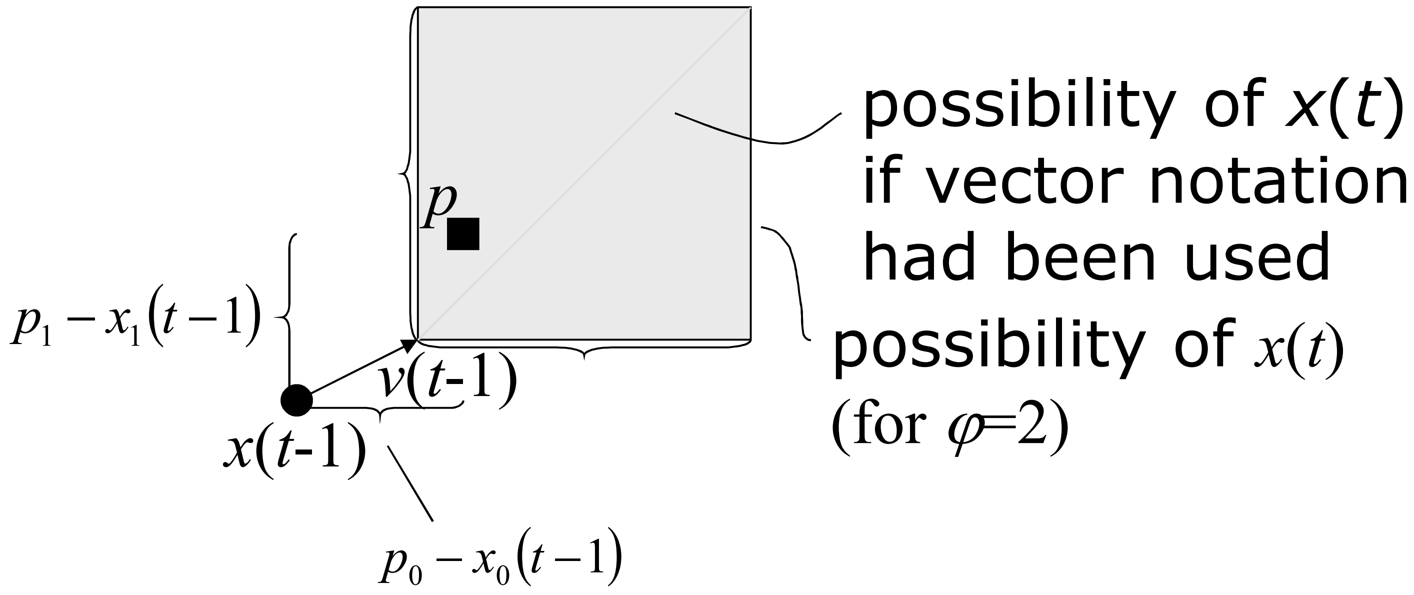
No interaction? That's not really a swarm, is it?

Olle Gällmo | olle.gallmo@it.uu.se

A pbest particle in action

$$v_d(t) = \underline{v_d(t-1) + \mathrm{U}(0,\varphi) * (p_d - x_d(t-1))} \qquad \forall d$$

$$x_d(t) = \underline{x_d(t-1) + v_d}(t)$$



Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

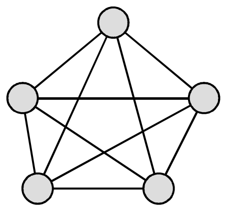
Global best (gbest)

- Global interaction

Best solution found so far by *any* particle

For all particles, i, and all dimensions, d:

$$v_{i,d}(t) = v_{i,d}(t-1) + \underbrace{U(0,\varphi_1) * (p_{i,d} - x_{i,d}(t-1))}_{\text{Cognitive component}} + \underbrace{U(0,\varphi_2) * (p_{g,d} - x_{i,d}(t-1))}_{\text{Social component}}$$



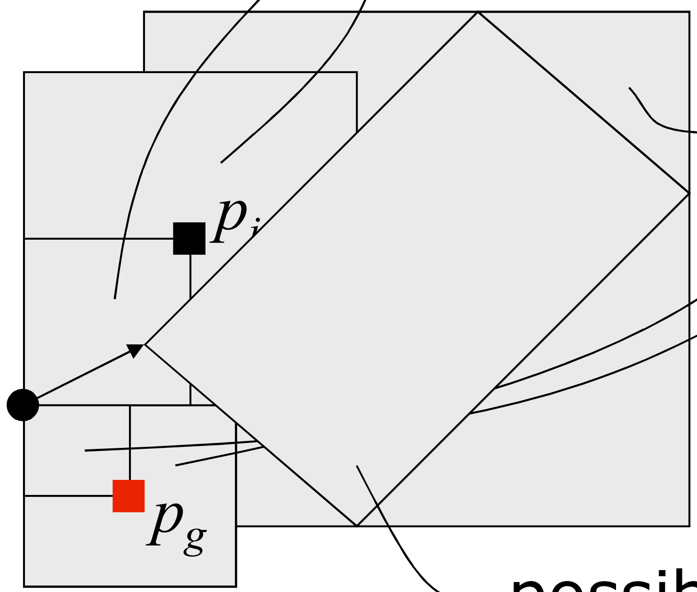
Star neighbourhood structure

To (immediately) know the global best is not very realistic, is it? (Yes, I know the Borg do)

UPPSALA UNIVERSITET

A gbest particle in action

$$v_{i,d}(t) = v_{i,d}(t-1) +$$
$$\mathrm{U}(0,\varphi_1)*(p_{i,d} - x_{i,d}(t-1)) + \mathrm{U}(0,\varphi_2)*(p_{g,d} - x_{i,d}(t-1))$$



possibility of $x(t)$, if $\varphi_1=\varphi_2=2$ and random numbers are drawn per element

possibility if random numbers are drawn per vector

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

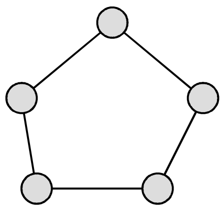
Local best (lbest)

- Local interaction

For all particles, i, and all dimensions, d:

$$v_{i,d}(t) = v_{i,d}(t-1) + \mathrm{U}(0,\varphi_1) * (p_{i,d} - x_{i,d}(t-1)) + \mathrm{U}(0,\varphi_2) * (p_{l,d} - x_{i,d}(t-1))$$

Best solution found so far by any particle *among i's neighbours* (in some structure)



Ring neighbourhood structure

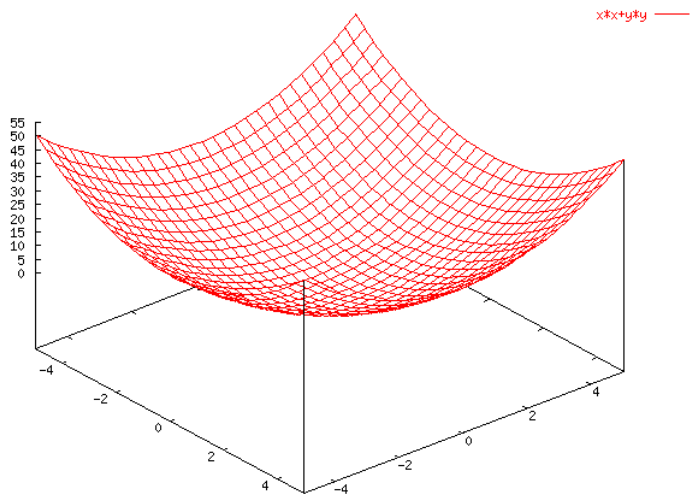
Nice, local, realistic, slower than gbest. Less risk of premature convergence

Olle Gällmo | olle.gallmo@it.uu.se

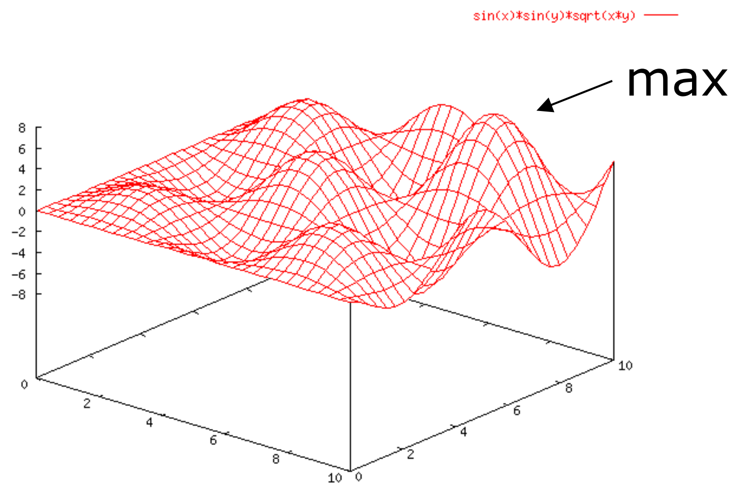
Simulation in 2D

Lbest with φ_1=1.8, φ_2=2.3

- Nhood: the 2 immediate neighbours
- V_{max} = range/25



Bowl



Alpine 2D

Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

Observations

- Usually requires a speed limit (V_{max})
- Actual velocity (v) usually close to V_{max}
- Discrete time ➔ velocity = step length
- Low accuracy close to global optimum
- Decaying V_{max}?
  - Imposes a time limit to reach the goal
- Inertia (meta-inertia, really)
- Constriction

Olle Gällmo | olle.gallmo@it.uu.se

(M. Clerc, 1999)

Constriction

- Constrict swarm with a factor K
  - to avoid divergence (”explosion”)
  - no longer need a speed limit
  - lowers speed around global optimum

$$v_{i,d}(t) = K * (v_{i,d}(t-1) + \mathrm{U}(0,\varphi_1) * (p_{i,d} - x_{i,d}(t-1)) + \mathrm{U}(0,\varphi_2) * (p_{l,d} - x_{i,d}(t-1)))$$

$$K = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}, \text{ where } \varphi = \varphi_1 + \varphi_2 > 4$$

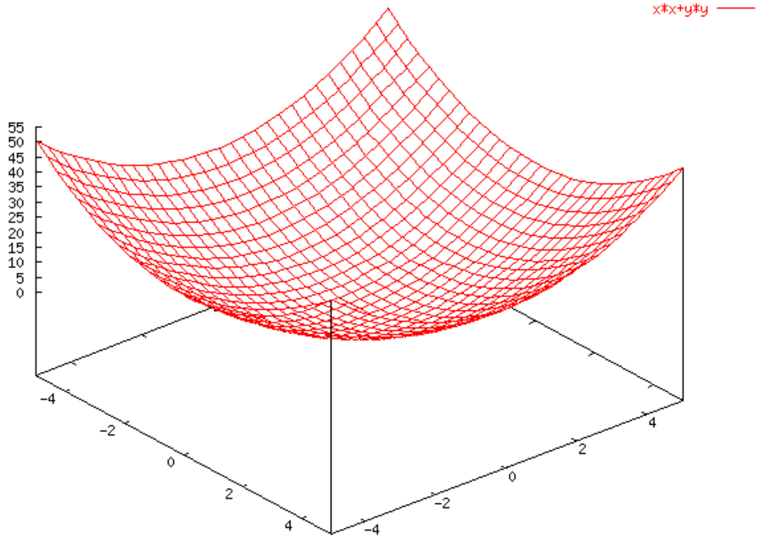
- K is a function of φ_1 and φ_2 only $\Rightarrow$ K is constant
  - yet it gives the swarm a converging behaviour, as if K was a decaying variable

Olle Gällmo | olle.gallmo@it.uu.se

Simulation with K

Lbest with φ_1=1.8, φ_2=2.3 $\Rightarrow$ K=0.7298

- Nhood: the 2 immediate neighbours
- No V_{max}



Binary particle swarms

- Velocities updated as before
- The positions:

$$x_{i,d} = \begin{cases} 1, \text{if } \mathrm{U}(0,1) < \text{logistic}(v_{i,d}) \\ 0, \text{if } \mathrm{U}(0,1) \geq \text{logistic}(v_{i,d}) \end{cases} \qquad \text{logistic}(v_{id}) = \frac{1}{1+e^{-v_{i,d}}}$$

- $V_{max} = \pm 4$
  - in order not to saturate the sigmoid
  - so that there is at least some probability (0.018) of a bit flipping

Olle Gällmo | olle.gallmo@it.uu.se

What makes PSO special?

- Its simplicity
- Adaptation operates on velocities
  - Most other methods operate on positions
  - Effect: PSO has a builtin momentum
  - Particles tend to hurdle past optima – an advantage, since the best positions are remembered anyway
- Few parameters to set, stable defaults
- Relatively easy to adapt swarm size
- Not very good for fine-tuning, though constriction helps

Olle Gällmo | olle.gallmo@it.uu.se

Notes on PSO

- Many publications are misleading on one important point:
  - The random numbers should be drawn *per element* (not per vector)
- Not really a swarm (though it behaves as one if V_{max} is small)
  - Particles don’t know other particles positions or velocities, only their personal bests
- The neighbourhood in *lbest* is structural (social)
  - Could it be topological (geographical)?

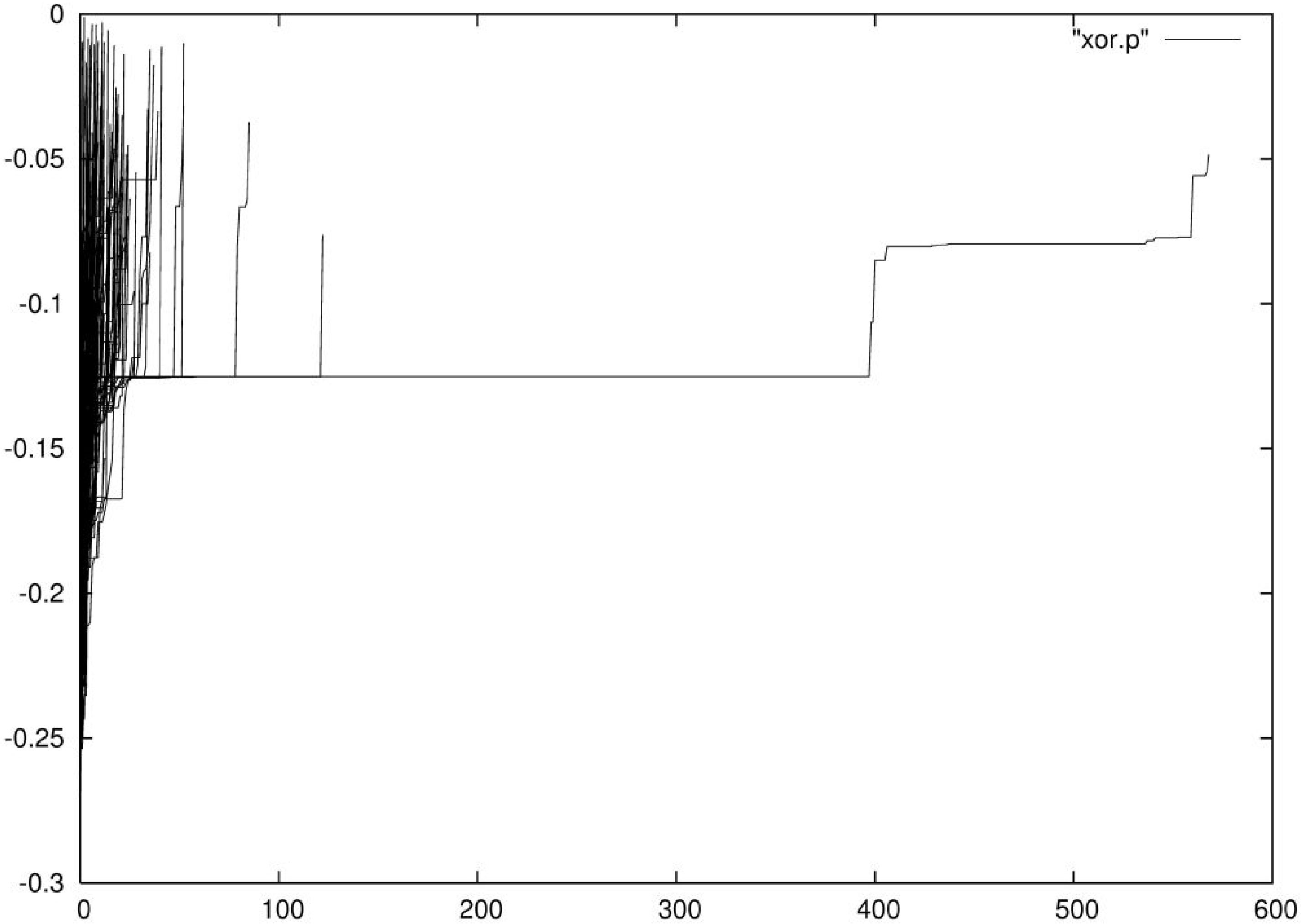
Olle Gällmo | olle.gallmo@it.uu.se

PSO and neural networks

- PSO can be used to train neural networks
- Each particle represents one network
  - x = a vector of all networks weights
  - also node type can be parameterized
- Hypothesis: This works better than EC
  - Crossover in EC may not be sound (for this)
    - Two individuals selected for crossover may have very little in common
  - Mutation in EC corresponds to moving the individual, but it is not directed, as in PSO
    - Both are random, but distribution depends on the situation in PSO, not so (usually) in EC
  - PSO population typically smaller than in EC
    - Speed advantage, since evaluation is a bottle-neck

Olle Gällmo | olle.gallmo@it.uu.se

MLP+PSO+XOR



Olle Gällmo | olle.gallmo@it.uu.se

UPPSALA UNIVERSITET

Grammatical Swarm

- Grammatical Evolution (GE) ...

BNF Grammar
<expr> ::= <const> | <expr><op><expr>
<const> ::= 0 | 1 | 2
<op> ::= + | -

Genome
23 | 4 | 17 | 42 | 62 | 10 | 6 | 22 | ...

$\Rightarrow$ 2+1

... where the integer string is trained by PSO instead of a genetic algorithm (GA)

Olle Gällmo | olle.gallmo@it.uu.se

Room for thought …

- High-dimensional spaces
- Bias along coordinate system axis
- Multiplying by vector instead of by element – effects in practice
- ”PSO on Ice” (PSO-GD)
- Dynamic neighbourhood, like GNG
- Parameter free PSO (Tribes)
- Multiple swarms and multiple goals

Olle Gällmo | olle.gallmo@it.uu.se

”Once again, nature has provided us with a technique for processing information that is at once elegant and versatile”

/ Kennedy & Eberhart -95

Olle Gällmo | olle.gallmo@it.uu.se