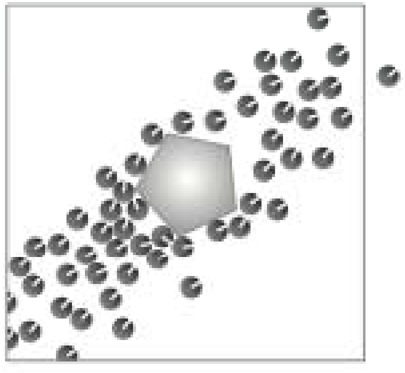
Swarm Intelligence

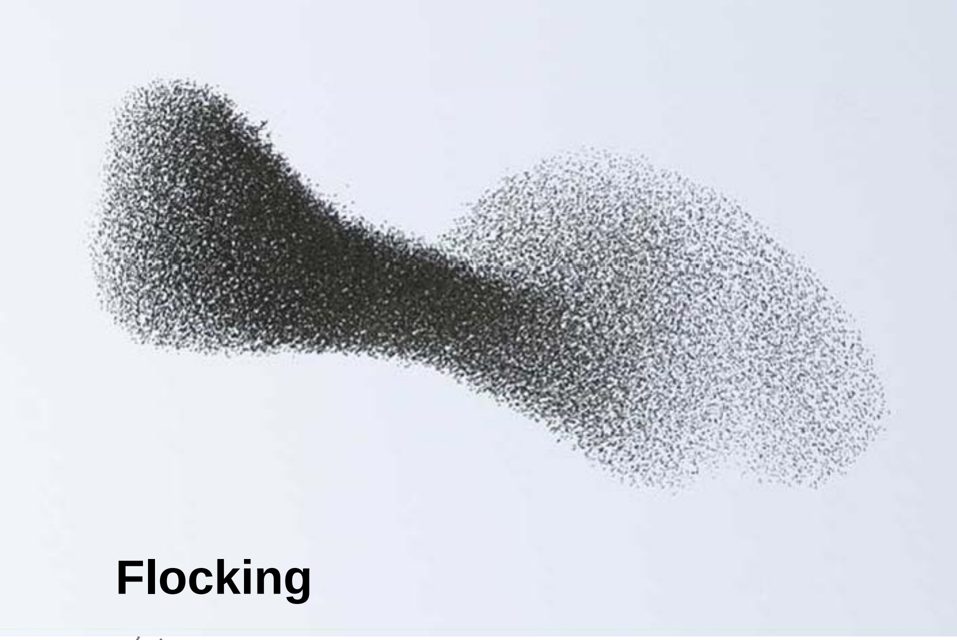


Emergent Collective Behavior

Some animal societies display coordinated and purposeful navigation of several individuals (from tens to thousands).

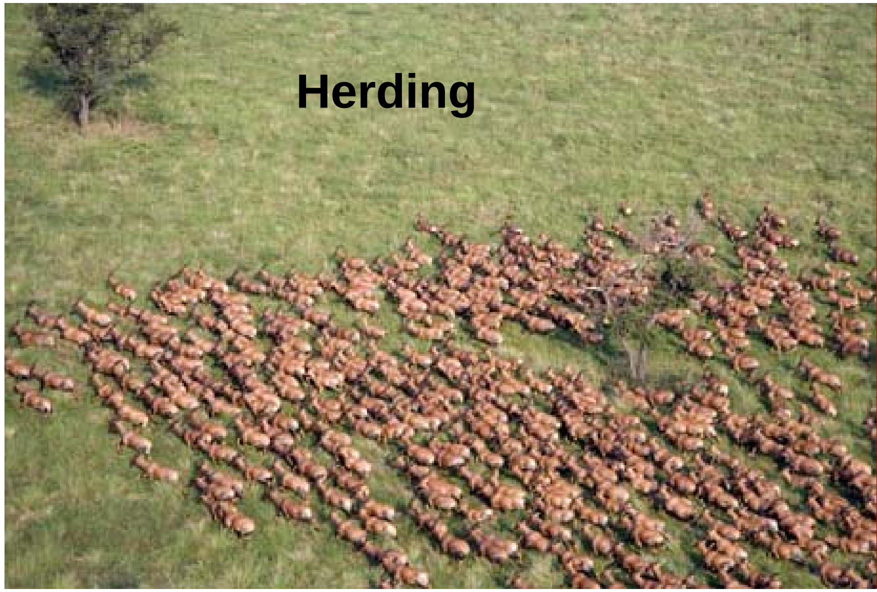
Each individual uses only local information about the presence of other individuals and of the environment.

There is no predefined group leader.



Emergent Collective Behavior

In some cases there is a leader and more restrictive rules on relative motion, but individuals still use local information to decide how to move.



Swarm Intelligence

Swarm Intelligence is the emergent collective intelligence of groups of simple individuals.

Main principles:

1) The swarm can solve complex problems that a single individual with simple abilities (computational or physical) could not solve.

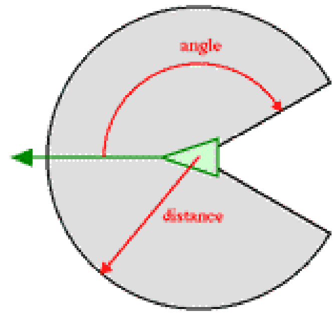
2) The swarm is composed of several individuals, some of which may be lost or make mistake, but its performance is not affected.

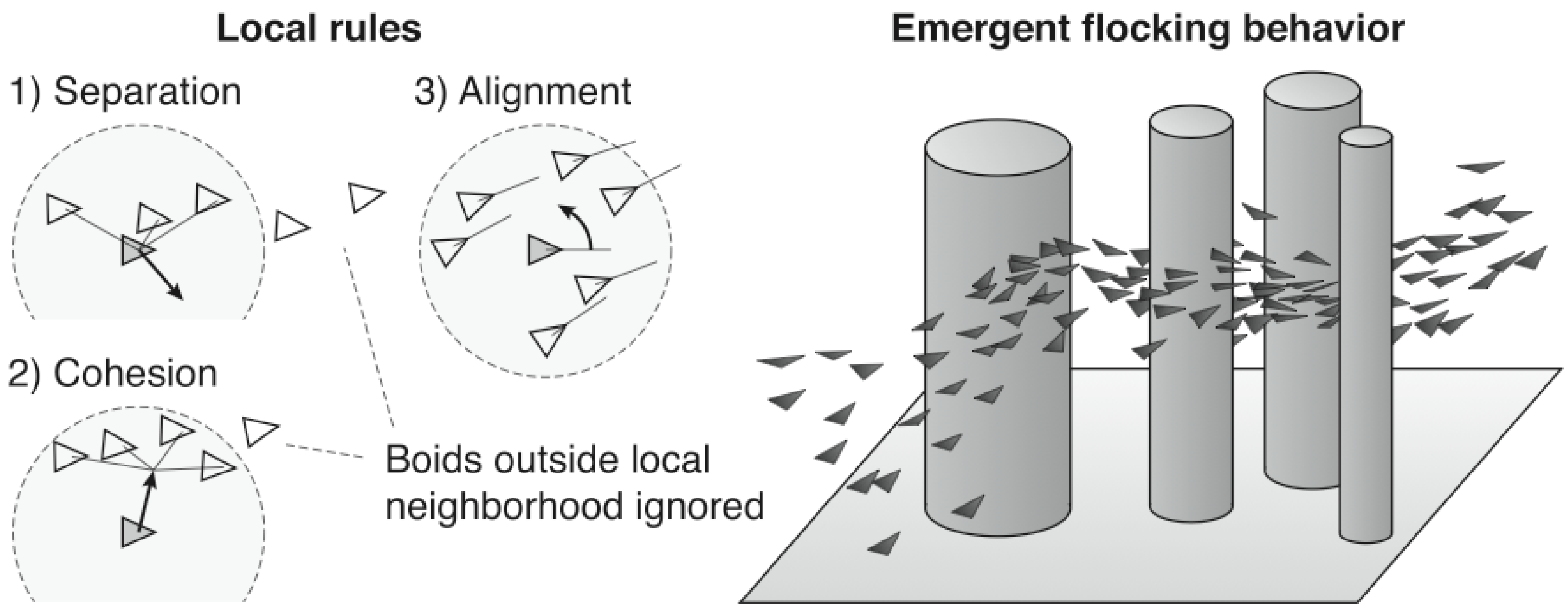
3) Individuals in a swarm have local sensory information, perform simple actions, have little/no memory; they do not know the global state of the swarm or its goal.

Coordinated navigation of swarms

Reynolds Flocking *(1987)*

Sensing: Boid perceives angle and distance of neighboring boids

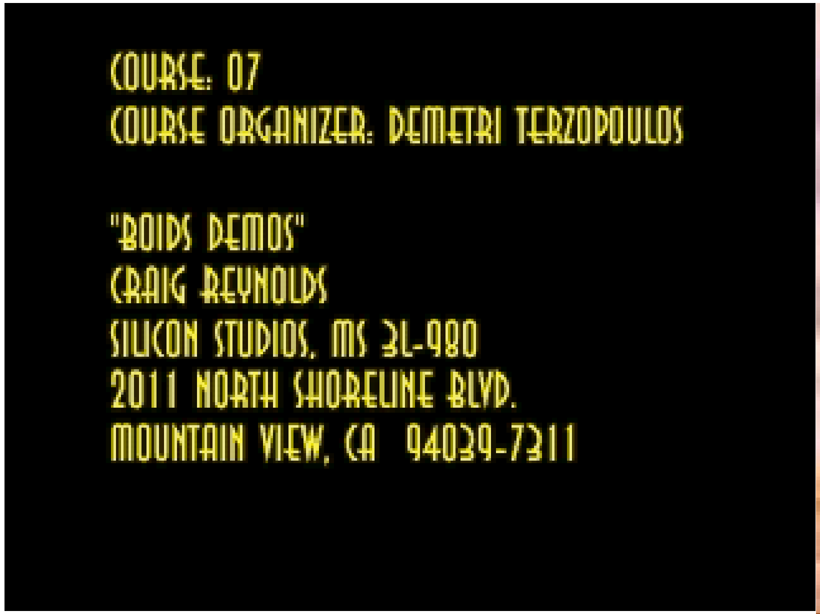




1. Separation: Boid maintains a given distance from other boids
2. Cohesion: Boid moves towards center of mass of neighboring boids
3. Alignment: Boid aligns its angle along those of neighboring boids

Examples of Character Animation

Emergent coordinated behavior. The approach is applicable to any type of animated characters in groups where behavior coordination is used.



The Lion King, 1994 (Walt Disney)

Challenges of Swarm Intelligence

Find individual behavioral rules that result in desired swarm behavior (reverse engineering).

Fortunately, the challenge may be addressed because the behavioral rules are supposed to be relatively simple. Often rules are hand-designed, sometimes are evolved.

Make sure the emergent behavior is stable.

Dynamical systems theory may help to characterize and predict swarm behavior because a swarm can be described as a system of elements with negative and positive interactions that moves in space and time. However, non-linear interactions are still hard to model.

Particle Swarm Optimization

Particle Swarm Optimization is an optimization algorithm inspired upon birds flocking to find the best food area.

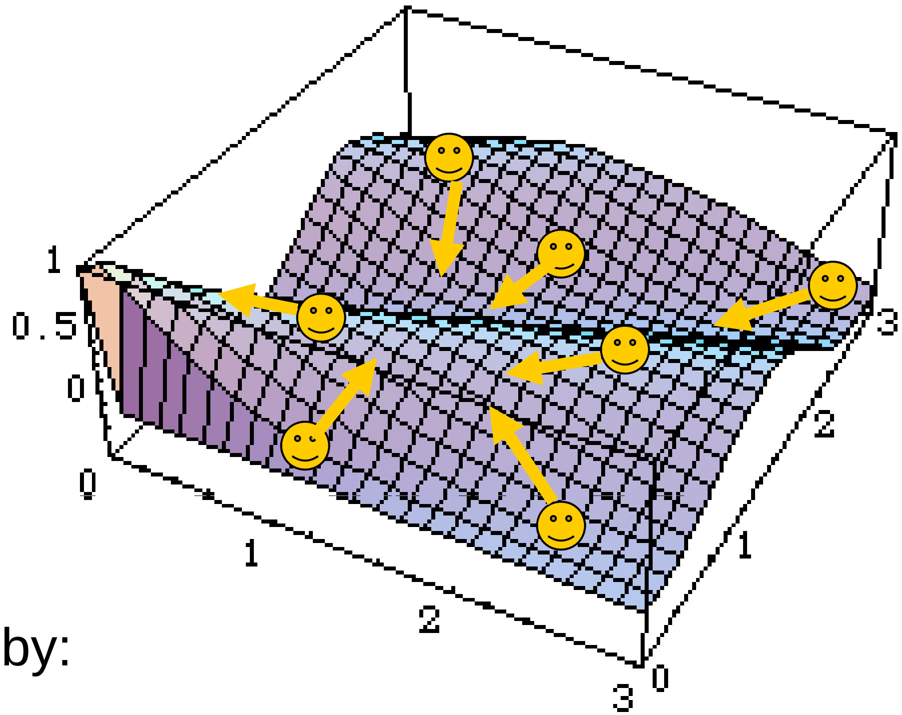
A caricature scenario:
The flock wants to find the area with the highest concentration of food (insects). Birds do not know where that area is, but each bird can shout to their neighbors how many insects are at its location. Birds also remember their own location where they found the highest concentration of food so far.

The flock is most likely to succeed when birds combine **three strategies**:

1) **Brave**: keep flying in the same direction
2) **Conservative**: fly back towards its own best previous position
3) **Swarm**: move towards its best neighbor

From Birds to Particles

The food concentration describes the search space of the optimization problem and the birds are the local solutions for that problem. They are called *particles* because they are very simple.

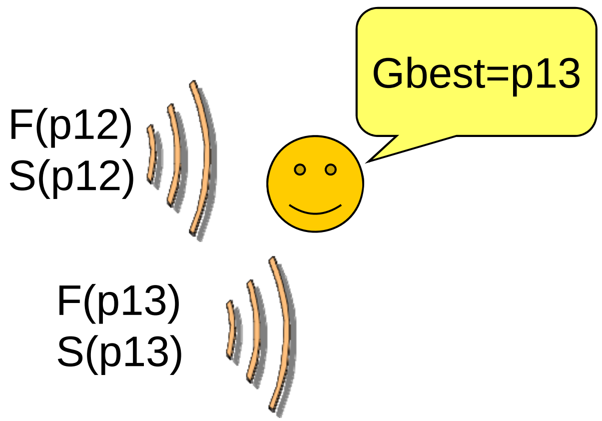


A particle **p** is described by:
s[] its position; e.g.: x, y
v[] its velocity; e.g. (for discrete case) angle and distance of next step
f[] its performance; e.g.: value of the function at its location

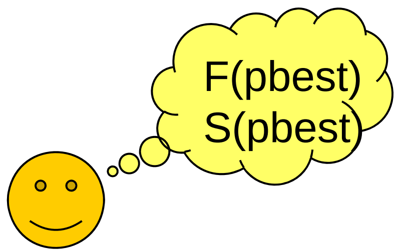
Particle's perception

A particle perceives performances and positions of neighboring particles.

It can also tell which is the best particle among its neighbors (gbest)

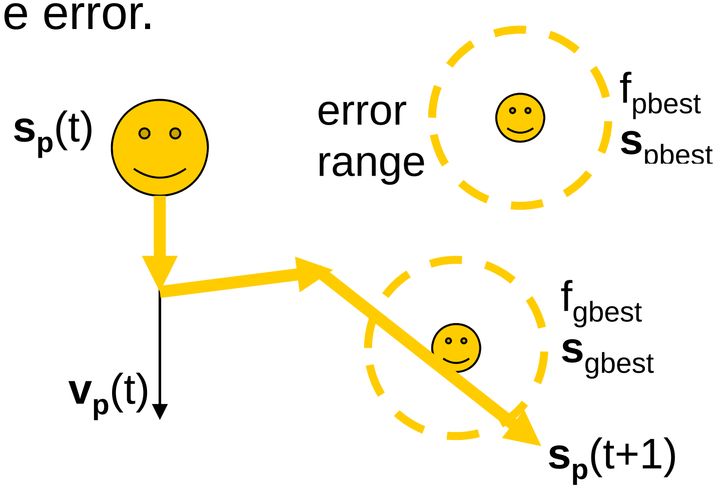


A particle remembers the position where it obtained the best performance so far (pbest)



Particle's actions

A particle computes the next position by taking into account a fraction of its current velocity **v**, the direction to its previous best location pbest, and the direction to the location of the best neighbor gbest. The movement towards other particles has some error.



$$\mathbf{v}_\mathbf{p}(t+1) = a \times \mathbf{v}_\mathbf{p}(t) + b \times R \times (\mathbf{s}_{pbest} - \mathbf{s}_\mathbf{p}(t)) + c \times R \times (\mathbf{s}_{gbest} - \mathbf{s}_\mathbf{p}(t))$$

$$\mathbf{s}_\mathbf{p}(t+1) = \mathbf{s}_\mathbf{p}(t) + \mathbf{v}_\mathbf{p}(t+1)$$

where a, b, c are learning constants between 0 and 1
R is a random number between 0 and 1

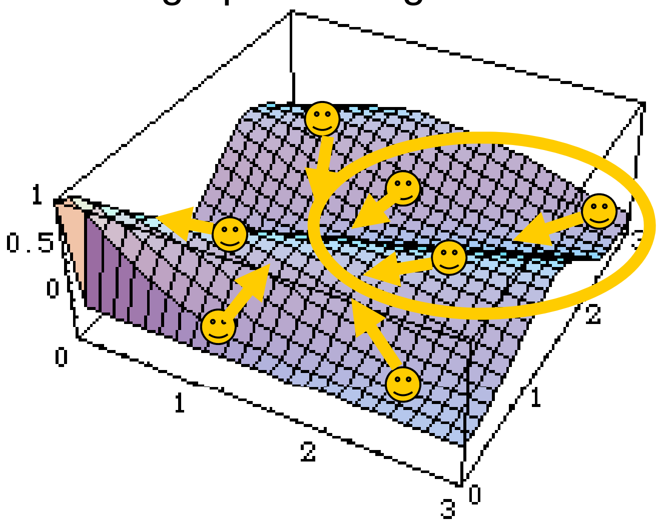
Initialization

Swarm size: Typically 20 particles for problems with dimensionality 2 - 200

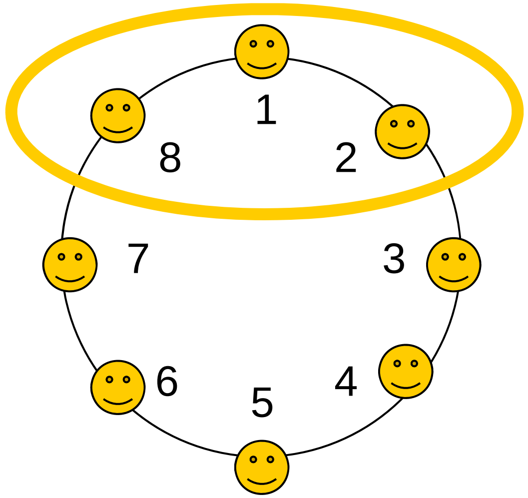
Initial position of each particle: Random

Neighborhood topology: Global, geographical or social (list based)

Geographical neighborhood



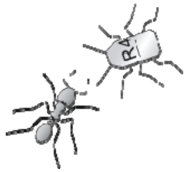
Social neighborhood



Neighborhood size: Typically 3 to 5

Set **max velocity** to v_{max}; if **v**(t+1) is larger, clip it to v_{max}

Iterate until best solution is found or no further improvement



PSO vs. Artificial Evolution

As in Artificial Evolution, PSO works with a population and some random factor to update solutions.

Contrary to Artificial Evolution, there is no generation change, no genome, and no competition among the individuals (rather cooperation)

A <u>major issue in PSO</u> is to transform the parameters of the problem to be solved so that it can be encoded and searched by particles

The best applications found so far include the large class of Traveling Salesman Problems and the optimization of neural network weights.

Reference: Kennedy and Eberhart (2001) Swarm Intelligence. Morgan Kauffman

Applet: http://www.projectcomputing.com/resources/psovis/

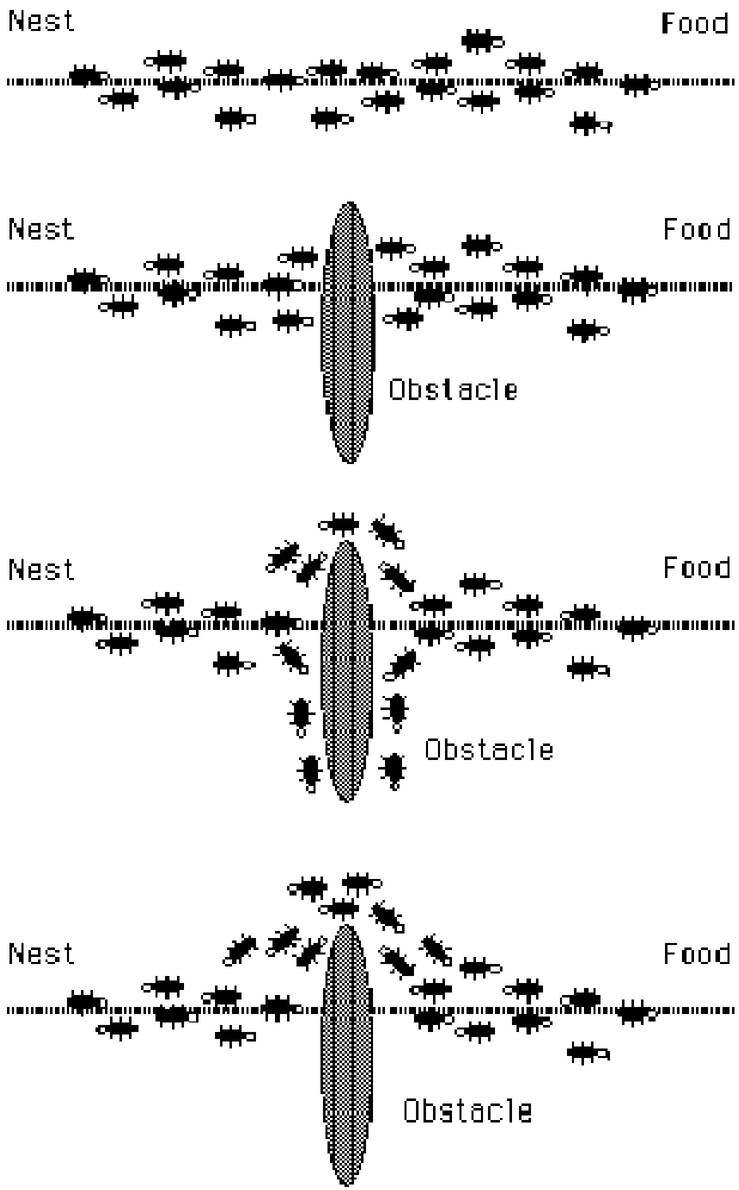
Ant trails

Stigmergy

The term indicates communication among individuals through modification of the environment.

For example, some ants leave a chemical (pheromone) trail behind to trace the path. *The chemical decays over time.*

This allows other ants to find the path between the food and the nest. It also allows ants to find the shortest path among alternative paths.

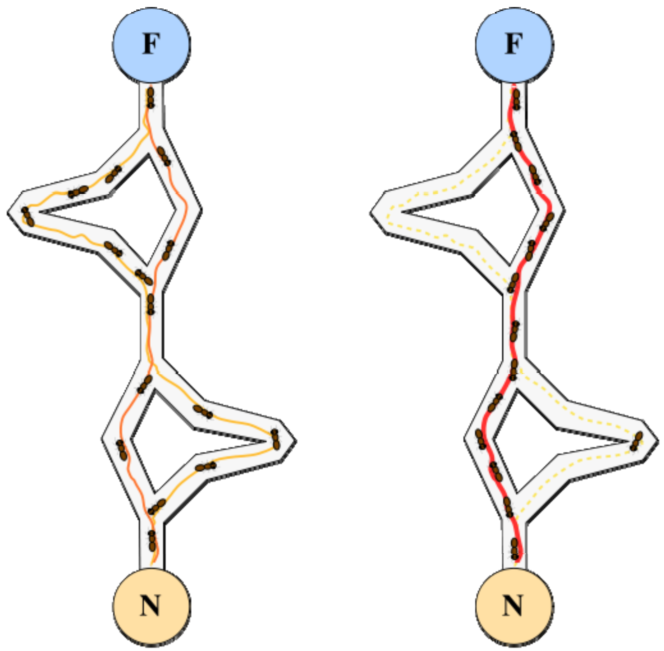


Finding the Shortest Path

1) As they move, ants deposit pheromone
2) Pheromone decays in time
3) Ants follow path with highest pheromone concentration
4) Without pheromone, equal probability of choosing short or long path

Shorter path allows higher number of passages and therefore pheromone level will be higher on shorter path.

Ants will increasingly tend to choose shorter path.



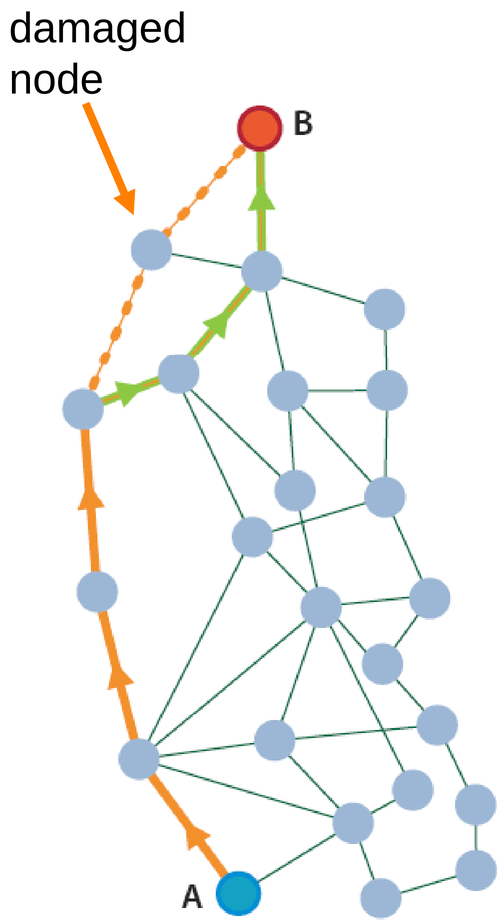
Goss et al. 1989, Deneubourg et al. 1990

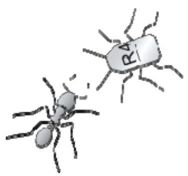
Ant Colony Optimization

Ant Colony Optimization is an algorithm developed by Dorigo et al. in 1994 inspired upon stigmergic communication to find the shortest path in a network.

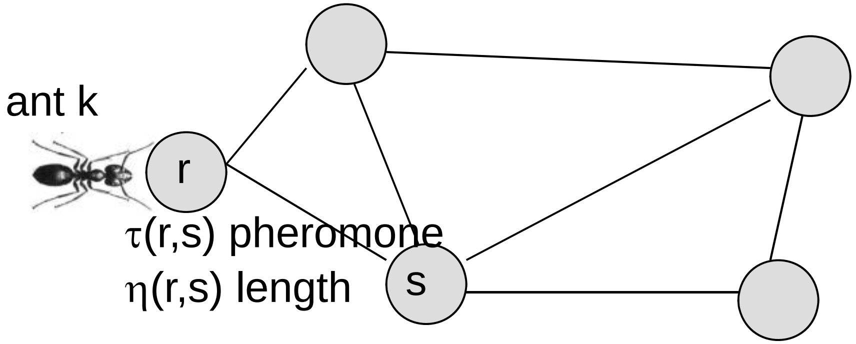
Typical examples are telephone, internet, and any problem that can be described as Travel Salesman Problem. Used/adopted by British Telecom, MCI Worldcom, Barilla, etc.

Advantage of algorithm is that, as ants do, it allows dynamic rerouting through shortest path if one node is broken. Most other algorithms instead assume that the network is static.





Ant Colony Optimization

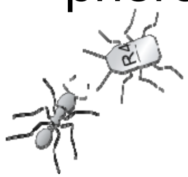


Each ant generates a complete tour of nodes using probabilistic transition rule encouraging choice of edge with high pheromone and short distance

Pheromone level on each edge is updated by considering evaporation and deposit by each ant

Pheromone levels only of edges traveled by best ant are increased in inverse proportion to length of path.

Result is that edges that belong to short tours receive greater amount of pheromone



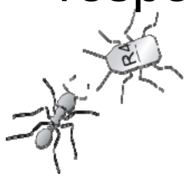
Transition Rule

Find a random number *q* between 0 and 1. If *q* is smaller than *q0*, then choose edge with largest amount of pheromone τ and shortest length η, otherwise use probabilistic rule:

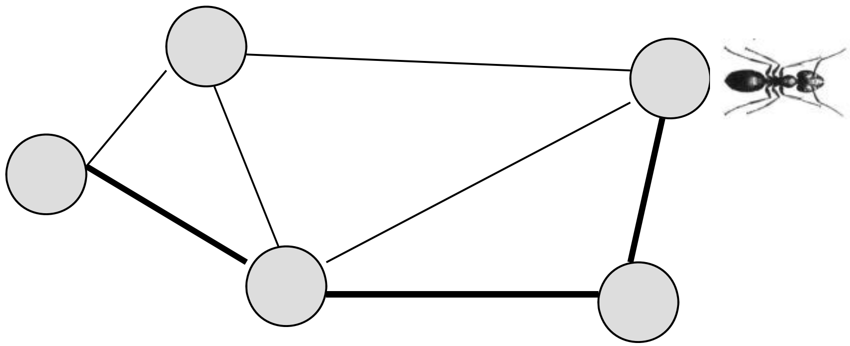
$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum\limits_{s \in J_k(r)} [\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}, & if \quad s \in J_k(r) \\ 0 & otherwise \end{cases}$$

Ant *k* sitting on city *r* moves to city *s* with probability proportional to amount of pheromone τ and length η of edge relative to all other cities connected to *r* that remain to be visited.

Choice of exponent β determines importance of edge length with respect to pheromone.



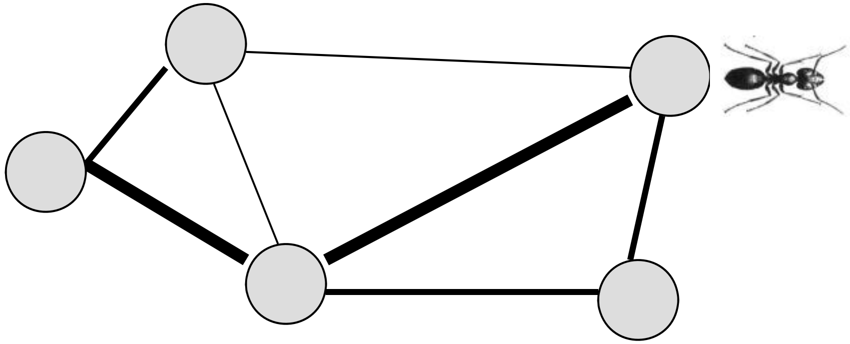
Pheromone Level Update: Local



$$\tau(r,s) \leftarrow (1-\rho)\cdot \tau(r,s) + \rho \cdot \tau 0$$

The pheromone level of each edge visited by an ant is decreased by a fraction (1- ρ) of its current level and increased by a fraction ρ of the initial level $\tau 0$.

Pheromone Level Update: Global



$$\tau(r,s) \leftarrow (1-\rho)\cdot \tau(r,s) + \rho \cdot L^{-1}$$

When all ants have completed their tours, the length L of the shortest tour is found and the pheromone levels of only the edges of this shortest path are updated in inverse proportion to the path length.

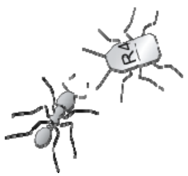
Initialization

Use approximately 100 ants

Distribute them on random nodes

Initial pheromone level is equal for all edges and inversely proportional to number of nodes times estimated length of optimal path

Initial pheromone level	$\tau 0 = \left(n \cdot L_{nn}\right)^{-1}$
Importance of length over pheromone	$\beta = 2$
Exploration threshold	$q0 = 0.9$
Pheromone update rate	$\rho = 0.1$



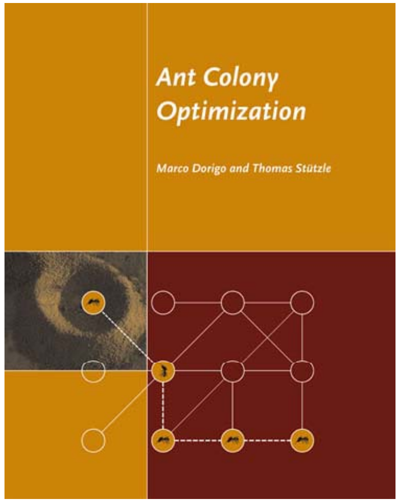
ACO Performance

Finds best solution on "small" problems (up to 30 cities)

Finds good solutions on large problems compared to other techniques

Finds best solution on large problems when coupled with other search techniques

Can operate on dynamic problems (e.g., node malfunctioning) that require fast rerouting



Dorigo and Stuetzle, 2005, MIT Press