Nature inspired metaheuristics

- Metaheuristics
- Swarm Intelligence
  - Ant Colony Optimization
  - Particle Swarm Optimization
  - Artificial Bees Colony

Metaheuristics

- Metaheuristic:

A **metaheuristic** is a heuristic method for solving a very general class of computational problems by combining user-given black-box procedures — usually heuristics themselves — in the hope of obtaining a more efficient or more robust procedure.

The name combines the Greek prefix "meta" ("beyond", here in the sense of "higher level") and "heuristic" (from ευρισκειν, *heuriskein*, "to find"). [Wikipedia]

- Nature inspired metaheuristics:

The ideas of these heuristics are inspired by the intelligent behavior of some living organisms.

Swarm intelligence

- Swarm intelligence = collection of intelligent techniques inspired by the collective behavior of some self-organizing systems
- The name was coined in 1989 by Gerardo Beni and Jing Wang in the context of control systems for robots
- The swarm intelligence techniques use sets of agents characterized by:
  - Simple functioning rules
  - Local interactions
  - No centralized control

Swarm intelligence

- Natural systems having such properties:
  - Ant colonies
  - Bee colonies
  - Bird swarms
  - Fish schools

- Such natural systems are models for techniques used in solving optimization and data analysis problems.

Imagini de la http://www.scs.carleton.ca/~arpwhite/courses/95590Y/notes/SI%20Lecture%203.pdf

Ant Systems

Inspiration: the behaviour of ant colonies when

- Searching for food -> solving an optimization problem = finding the shortest route between the food source and the nest

- Organizing the nest -> solving a data clustering problem = organizing the data items based on similarity between them

Key elements:

- The ants communicate indirectly by using some chemical substances called pheromones; this communication process is called stigmergy (useful in solving optimization problems)

- The ants belonging to the same nest recognize each other by odour (useful in data clustering)

Ant systems

Illustration of stigmergy: the experiment of the double bridge [Deneubourg, 1990]

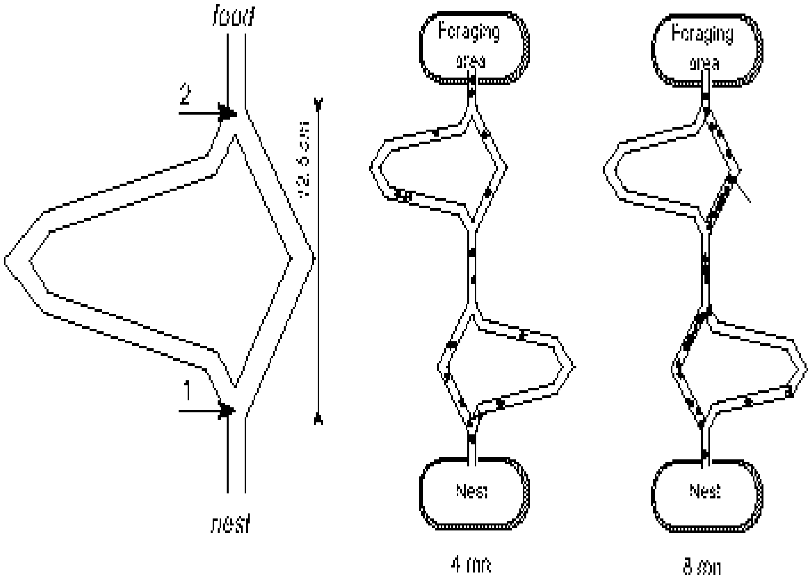
Ant species: Argentine

- There are two accessing routes between the food and the nest

- Initially the ants choose randomly the route

- When an ant goes from the food to the nest it releases pheromones on the route

- The shorter route will soon have a higher pheromone concentration

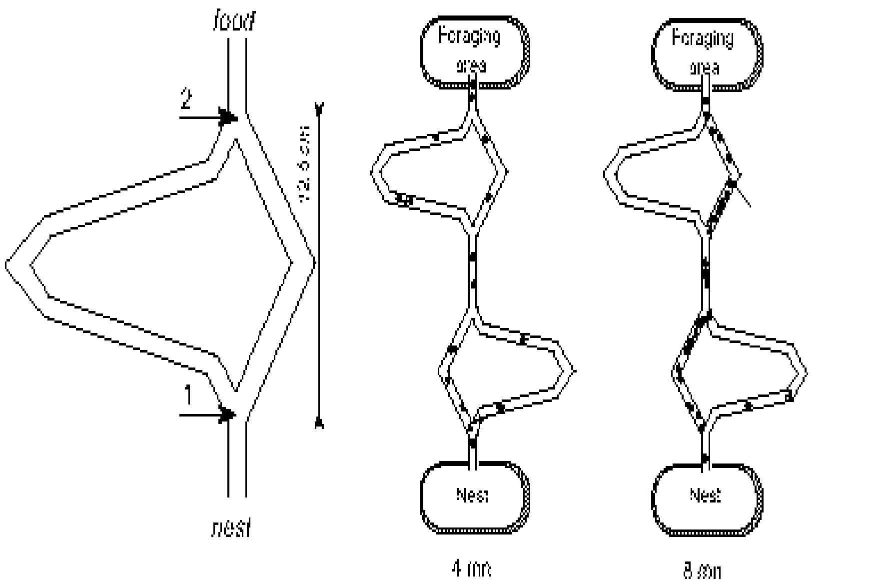


Ant systems

Illustration of stigmergy: the experiment of the double bridge

- If the concentration of pheromones differ between two paths the ants will prefer the path with a higher concentration.

- Thus, in time more and more ants will choose the route with a higher concentration (which is the shorter route) and the concentration will be increased. This is a positive feedback phenomenon.



Ant Systems

Illustration of stigmergy: the experiment of the double bridge

- The pheromone concentration also can decrease in time because of an evaporation phenomenon

- The evaporation is useful especially in the case of dynamic environments (there are some changes in the environment)

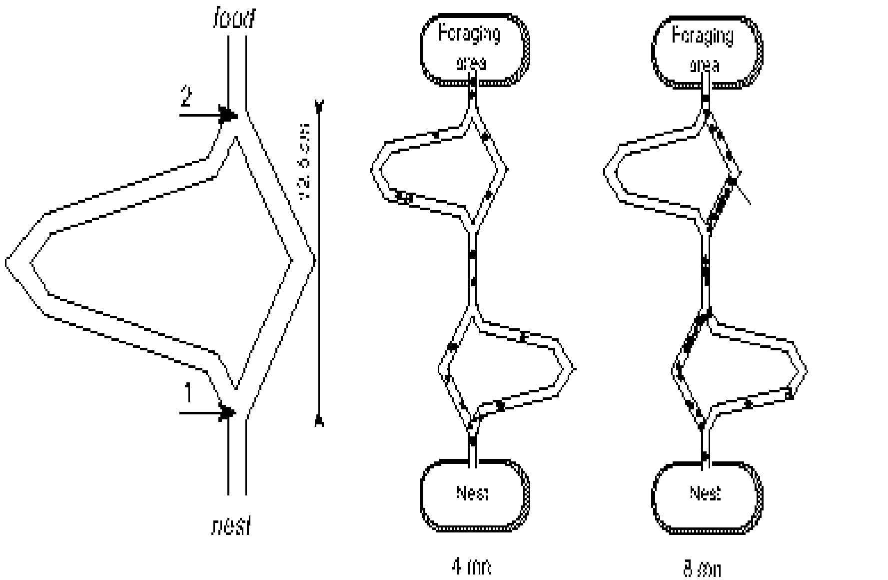


Illustration: http://www.nightlab.ch/downloads.php

Ant Systems

Solving an optimization problem – Ant Colony Optimization

Idea: the solution of the problem is constructed using a set of artificial ants (agents) which indirectly communicate information concerning the quality of the solution

Example: travelling salesman problem

Input: labelled graph specifying the direct connections between towns (nodes in graph) and their costs

Output: a visiting order of towns such that the total cost is minimal

Ant Colony Optimization

ACO for travelling salesman problem [Dorigo, 1992]:

- There is a set of ants which are involved in an iterative process

- At each iteration each ant constructs a route by visiting all nodes of the graphs. When it constructs a route each ant follows the rules:

  - It does not visit twice the same node
  - The decision to chose the next node to visit is probabilistically taken by using both information related to the cost of the corresponding edge and the concentration of pheromone stored on that edge.

- After all ants constructed their tours the pheromone concentration is updated by simulating the evaporation process and by rewarding the edges which belong to tours having a small total cost.

Ant Colony Optimization

General structure of the algorithm

```
Initialize the pheromone concentrations tau(i,j) for all
   edges (i,j)
FOR t:=1,tmax DO
  FOR k:=1,m DO   // each ant constructs a tour
      i1(k):=1
      FOR p:=2,n DO
           chose ip(k) using the probability Pk(ip-1,ip)
  compute the cost of all tours
  update the pheromone concentrations
```

Notations:

tmax = number of iterations; m=number of agents (ants);

i(k) = tour constructed by ant k

p = node index

P = transition probability, tau = pheromone concentration

Ant Colony Optimization

Variants:

Algorithm	*Authors*	*Year*
Ant System (AS)	Dorigo et al.	1991
Elitist AS	Dorigo et al.	1992
Ant-Q	Gambardella & Dorigo	1995
Ant Colony System	Dorigo & Gambardella	1996
$\mathcal{MAX}$–$\mathcal{MIN}$ AS	Stützle & Hoos	1996
Rank-based AS	Bullnheimer et al.	1997
ANTS	Maniezzo	1999
BWAS	Cordón et al.	2000
Hyper-cube AS	Blum et al.	2001

Rmk: the variants differ mainly with respect to the computation of the transition probability and with respect to the rule for updating the pheromone concentration

Ant Colony Optimization

Original variant for TSP (AS - Ant Systems)

Solution encoding: $(i_1,i_2,\ldots,i_n)$ is a permutation of the set of nodes indices

Transition probabilities

(the ant k moves at iteration t from node i to node j)

$$P_{ij}^{k}(t) = \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{l\in A(i,k)}\tau_{il}^{\alpha}(t)\eta_{il}^{\beta}(t)}$$

A(i,k) = list of nodes unvisited by ant k which are connected to node i

α and β are constants controlling the relative weight of each factor (pheromone concentration vs. heuristic factor)

The pheromone concentration on edge (i,l)

Heuristic factor related with the cost of edge (i,l);
Usually it is 1/cost(I,j)

Ant Colony Optimization

Original variant for TSP (AS - Ant Systems)

Pheromone concentration updating (at the end of each iteration)

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta_{ij}^{k}$$

$$\Delta_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if ant k used the edge (i, j)} \\ 0 & \text{otherwise} \end{cases}$$

Remark:

Another variant could be based on adjusting the pheromone levels using only the best tour T* constructed during the last iteration:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta_{ij}^{*}$$

$$\Delta_{ij}^{*} = \begin{cases} \dfrac{Q}{L^{*}} & \text{if } T^{*} \text{ contains (i, j)} \\ 0 & \text{otherwise} \end{cases}$$

Notations:

ρ = evaporation rate

Q>0 = constant

L_k = cost of last tour constructed by ant k

Other variants

Particularities of other variants:

Max-Min Ant System (MMAS):

- the pheromone concentration is limited to values in a given interval
- the pheromone concentration is increased only for edges which belong to the best tour found during the previous iteration (see remark on the previous slide)

Ant Colony System (ACS)

- besides the global updating of pheromone concentration used in MMAS it also used a local updating of pheromone concentration which is applied any time an arc is visited

$$\tau_{ij} = (1-\varphi)\cdot\tau_{ij} + \varphi\cdot\tau_0\ ,$$

Initial value of the concentration

Ant Systems

Applications

Problem type	*Problem name*	*Authors*	*Year*
Routing	Traveling salesman	Dorigo et al.	1991, 1996
		Dorigo & Gambardella	1997
		Stützle & Hoos	1997, 2000
	Vehicle routing	Gambardella et al.	1999
		Reimann et al.	2004
	Sequential ordering	Gambardella & Dorigo	2000
Assignment	Quadratic assignment	Stützle & Hoos	2000
		Maniezzo	1999
	Course timetabling	Socha et al.	2002, 2003
	Graph coloring	Costa & Hertz	1997
Scheduling	Project scheduling	Merkle et al.	2002
	Total weighted tardiness	den Besten et al.	2000
	Total weighted tardiness	Merkle & Middendorf	2000
	Open shop	Blum	2005
Subset	Set covering	Lessing et al.	2004
	l-cardinality trees	Blum & Blesa	2005
	Multiple knapsack	Leguizamón & Michalewicz	1999
	Maximum clique	Fenet & Solnon	2003
Other	Constraint satisfaction	Solnon	2000, 2002
	Classification rules	Parpinelli et al.	2002
		Martens et al.	2006
	Bayesian networks	Campos, Fernández-Luna,	2002
	Protein folding	Shmygelska & Hoos	2005
	Docking	Korb et al.	2006

Ant Systems

Applications in real problems:

- Routing problems (telecommunication networks, vehicles)
- Dynamic optimization problems
- Task scheduling

Companies which applied ant algorithms in solving real problems:

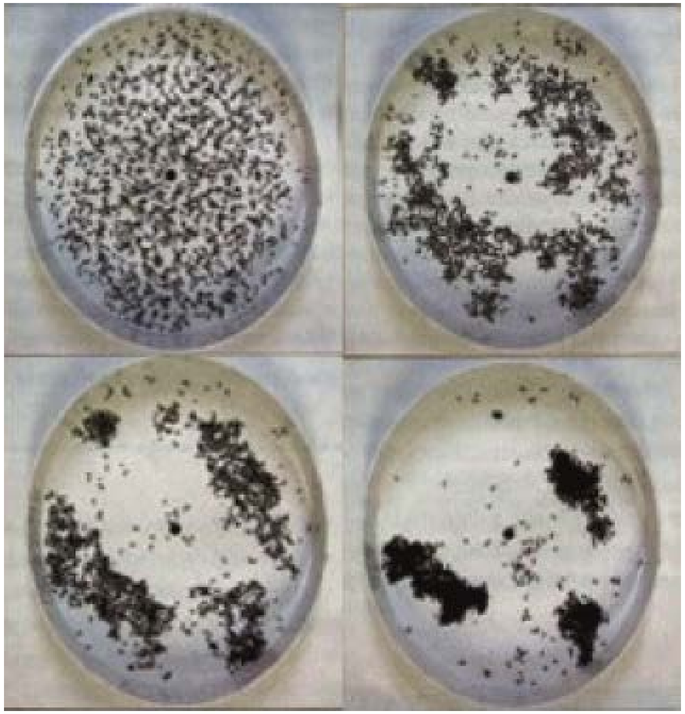
www.eurobios.com (routing/schedule of airplane flights, supply chain networks)
www.antoptima.com (vehicle routing)

Ant Systems

Applications in data analysis. It uses as inspiration

- The process by which the ants organizes the food or the bodies of dead ants (Lumer &Faieta, 1994)

- The process in which the ants identify ants belonging to other species (AntClust – Labroche, 2002)



Ant clustering

AntClust – clustering algorithm [Labroche, 2002]

- AntClust [Labroche et al., 2002] simulates the “colonial closure” phenomenon in ants colonies:
  - It is inspired by the chemical odors used by ants to make the difference between nestmates and intruders
  - The interaction between ants is modeled by so-called meetings when two ants confront their odors

Ant Colony

- Ant
- Nest (ants with similar odors)
- Odor template
- Meeting between two ants
- Nest creation
- Ant migration between nests
- Ant elimination from a nest

Clustering process

- Data
- Cluster (class of similar data)
- Similarity threshold
- Comparison between two data
- Cluster initiation
- Data transfer from a cluster to another one
- Data elimination from a cluster

Ant Clustering

- To cluster m data there are used m artificial ants, each one being characterized by:
  - An associated data, x
  - A label identifying the cluster, L
  - A similarity threshold, T
  - A parameter counting the meetings an ant participates to, A
  - A parameter measuring the ant's perception of its nest size, M
  - A parameter measuring the ant's perception of the acceptance degree by the other members in its nest, M^+

- Structure of AntClust
  - Threshold's learning phase
  - Meetings phase
  - Clusters refining phase

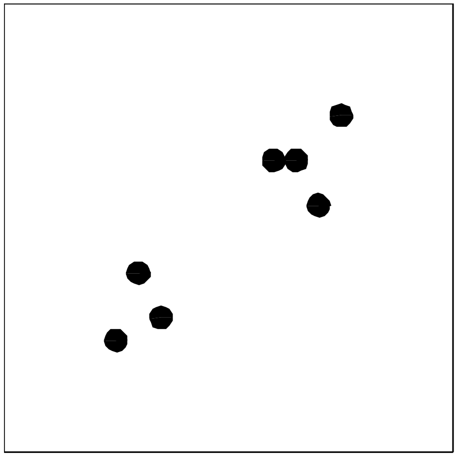
Ant Clustering

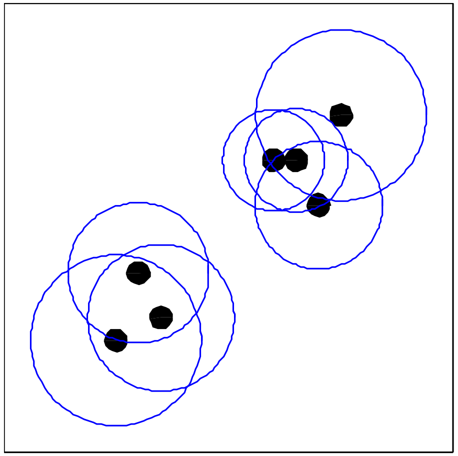
- Threshold's learning phase:
  - For each ant the threshold T is estimated based on the maximum and averaged similarity between its corresponding data and other data

$$T_i = \frac{\max_j(S(i,j)) + \operatorname{avg}_j(S(i,j))}{2}$$

$$S(i,j) = \frac{1}{n}\sum_{k=1}^{n}\left(1 - \frac{|x_i^k - x_j^k|}{\max x^k - \min x^k}\right)$$

Data





Areas of similarity illustration

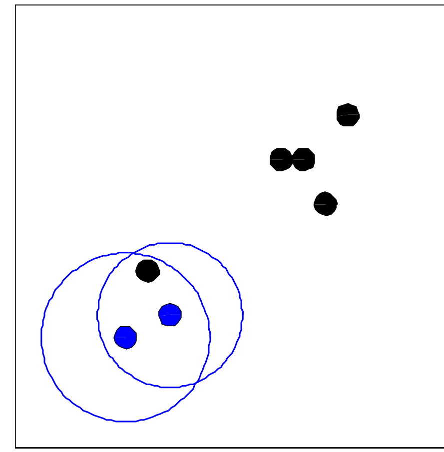
Ant Clustering

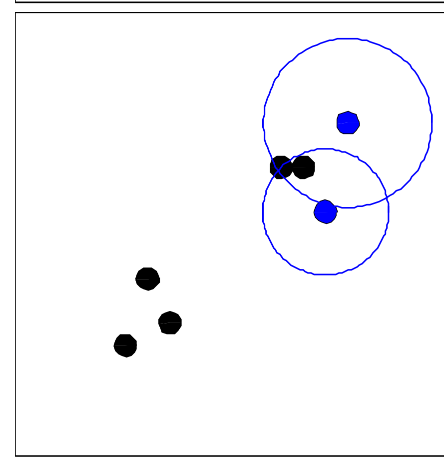
- Random meetings phase:
  - Pairs of ants are randomly selected for k_M times
  - When ant i meets ant j, the similarity S(i,j) is computed and the following analysis is made:

```
If S(i,j)>Ti and S(i,j)>Tj
then the ants accept each other
else they reject each other
```

  - A set of rules is applied depending on the acceptance or rejection; the effect of these rules consists of modifications of the labels and of the perception parameters

Acceptance situation





Rejection situation

Ant Clustering

- Acceptance rules:

Rule 1:

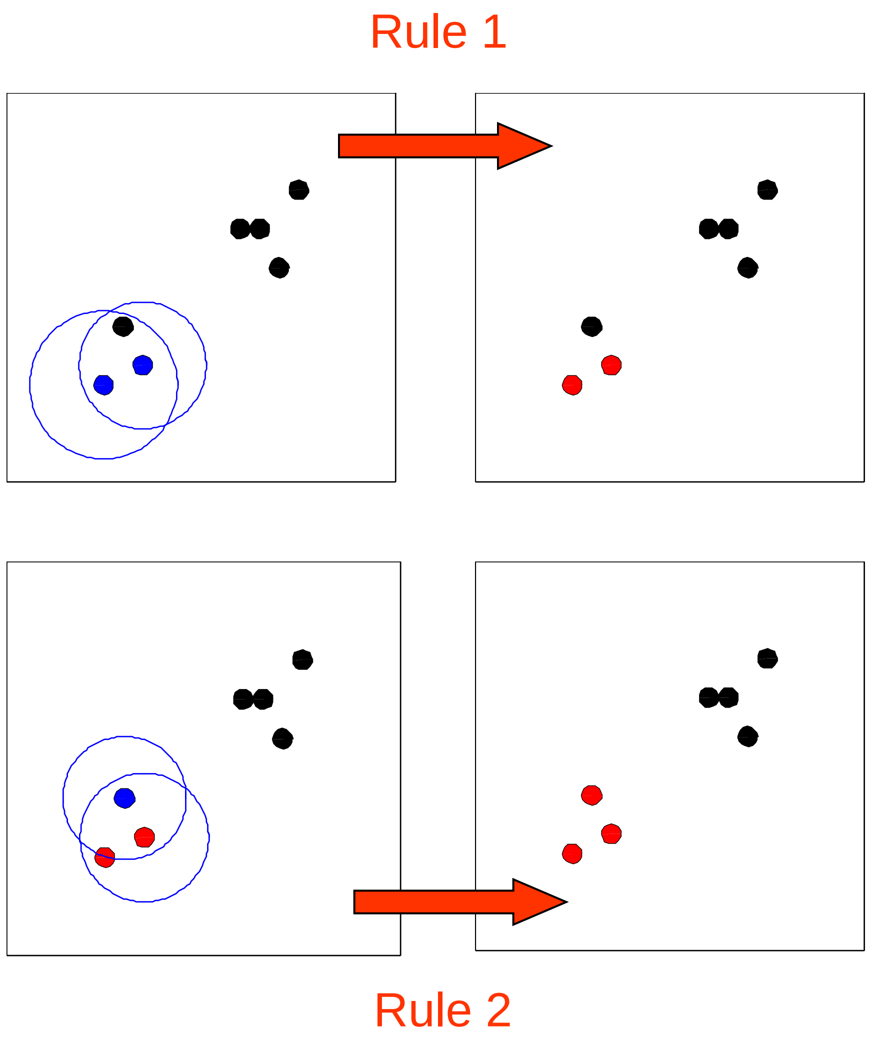
`If two unlabeled ants meet`

`then they will create a new nest`

Rule 2:

`If an unlabeled ant meets a labeled one`

`then it is included in the same nest`



Ant Clustering

- Acceptance rules:

Rule 3:

If two ants belonging to the same nest meet

then their perception parameters, M and M^+ are increased

Rule 5:

If two ants belonging to different nests meet

then the ant having the lower M is attracted into the nest of the other ant and their M parameters are decreased

Increase function

$$\mathrm{inc}(v) = (1-\alpha)v + \alpha$$

Decrease function

$$\mathrm{dec}(v) = (1-\alpha)v$$

$\alpha \in (0,1)$ is a parameter

M and M^+ belongs to [0,1)

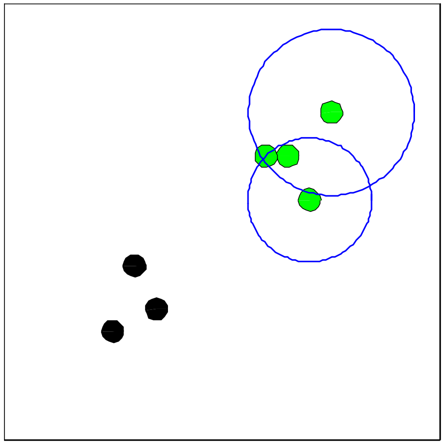
Ant Clustering

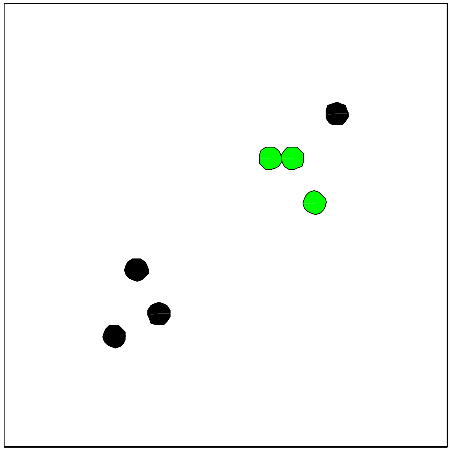
- Rejection rule:

Rule 4:

```
If two rejecting ants which belong
   to the same nest meet
then
```

- the ant having the lower M^+ is eliminated from the nest and its parameters are reset
- the M parameter of the other ant is increased
- the M^+ parameter of the other ant is decreased





Ant Clustering

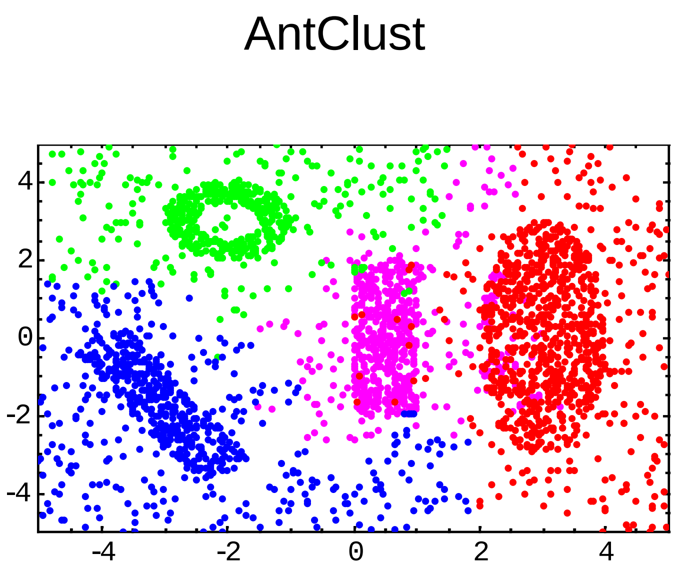
❑ The Algorithm

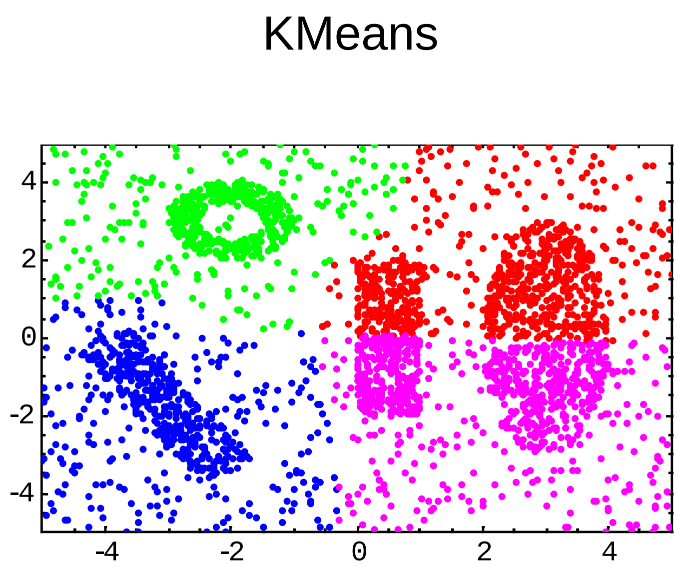
Algorithm 1 AntClust algorithm

for all $i \in \{1, \ldots, m\}$ **do**
  $L_i := 0;\ A_i := 0;\ M_i := 0;\ M_i^+ := 0;$
end for
{Threshold learning:}
for all $i \in \{1, \ldots, m\}$ **do**
  sample k_T ants and compute
  $max\{S(i, \cdot)\}$ and $\langle S(i, \cdot)\rangle$;
  $T_i := (max\{S(i, \cdot)\} + \langle S(i, \cdot)\rangle)/2;$
end for
{Random meetings:}
for all $k \in \{1, \ldots, k_M\}$ **do**
  Select a random pair (i, j)
  Increase the age: $A_i := A_i + 1;\ A_j := A_j + 1;$
  Compute $S(i, j)$
  Apply the rules R1-R5
end for
{Clusters refining:}
for all identified clusters **do**
  compute $N_{cluster}$ (the ratio of data belonging to the cluster) and
  $\langle M^+\rangle$ (averaged value of M^+ for all ants in the cluster)
  compute the acceptance probability
  $P_a = \alpha\langle M^+\rangle + (1 - \alpha)N_{cluster}$
  if $P_a < \theta_r$ **then**
    remove the cluster (all its elements are reset)
  end if
end for
for all ants having $M^+ < \theta_a$ **do**
  assign the ant to the cluster of the most similar ant, j, which belongs to a nest ($L_j \neq 0$) and has a high enough acceptance degree ($M_j^+ > \theta_a$)
end for

Ant Clustering

- Example





Particle Swarm Optimization

- It has been designed by James Kennedy şi Russell Eberhart for nonlinear function optimization (1995)

- Inspiration:
  - The behaviour of bird swarms, fish schools
  - The birds are considered similar to some particles which “flies” in the search space to identify the optimum

- Biblio: http://www.particleswarm.info/

Particle Swarm Optimization

Idea:

- Use a set of “particles” placed in the search space
- Each particle is characterized by:
  - Its position
  - Its velocity
  - The best position found so far
- The particles position change at each iteration based on the
  - Best position found by the particle (personal best)
  - Best position identified by the swarm (global best)

General structure:

Initialization of particle positions

REPEAT

compute the velocities

update the positions

evaluate new positions

update the local and global memory

UNTIL <stopping condition>

Ilustrare: http://www.projectcomputing.com/resources/psovis/index.html

Particle Swarm Optimization

- Updating the particle velocities and positions

$$v_i^j(t+1) = v_i^j(t) + c \cdot r_1(t)(p_i^j(t) - x_i^j(t)) + c \cdot r_2(t)(p_b^j(t) - x_i^j(t))$$

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t+1)$$

Particle Swarm Optimization

- Updating the particle velocities and positions

Best position of the swarm

Best position of particle i

$$v_i^j(t+1) = v_i^j(t) + c \cdot r_1(t)(p_i^j(t) - x_i^j(t)) + c \cdot r_2(t)(p_b^j(t) - x_i^j(t))$$

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t+1)$$

Random values In (0,1)

j^{th} component of the velocity of particle i at iteration (t+1)

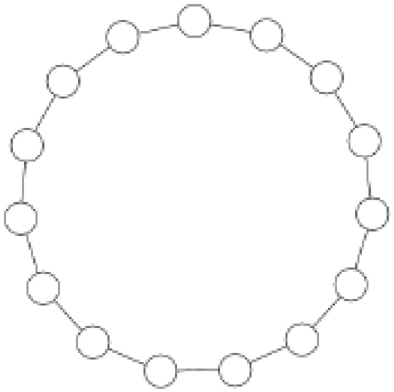
j^{th} component of the position of particle i at iteration t

Particle Swarm Optimization

- Variants
  - Use an inertia factor (w) and a constriction factor in order to limit the velocity value (gamma)

$$v_i^j(t+1) = \gamma(wv_i^j(t) + c_1r_1(t)(p_i^j(t) - x_i^j(t)) + c_2r_2(t)(p_b^j(t) - x_i^j(t)))$$

  - Use instead of a global best position the best position in a neighborhood of the current element
    - Example: circular topology



Artificial Bee Colony

- Artificial Bee Colony (ABC) [Karaboga, 2005]
  http://mf.erciyes.edu.tr/abc/links.htm

- Inspiration: intelligent behavior of bees when they search for honey

- Use a population of “bees” consisting of three types of bees

  - Employed bees (they are already placed in a source of honey)
  - Observing (onlooker) bees (they collect information from employed bees)
  - Scouters (they randomly search for new sources of honey)

Artificial Bee Colony

- Employed bees
  - They are associated with a particular food source which they are currently exploiting or are “employed” at.
  - They carry with them information about this particular source, its distance and direction from the nest, the profitability of the source and share this information with a certain probability.
- Observing bees (onlookers):
  - They wait in their nest and select a food source through the information shared by employed bees
- Scout bees
  - They search the environment surrounding the nest for new food sources

Artificial Bee Colony

- Step 1: Random initialization of positions of employed bees
- Step 2. While not stopping condition:
  - The employed bees send information about the quality of their location to onlookers; each onlooker receives information from several employed bees; the selection is based on a probability distribution computed by using the fitness of the analyzed locations.
  - The employed bees explores the neighborhood of their location and moves in a different one if this last one is better; if the bee cannot find a better location in a given number of steps then it is randomly relocated (e.g. to a position provided by a scout).
  - The scout bees change randomly their position.

Artificial Bee Colony

Details:

- Notations NB = number of employed bees, NO = number of onlookers,
  f = fitness function, n = problem size
- The distribution probability of the new location selected by onlookers:

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^{NB} f(x_j)}$$

  - The choice of the new position by an onlooker can be implemented by using the roulette technique
  - The employed bees are relocated using the following rule (k is the index of a random employed bee, ϕ_{ij} is a random value in [-1,1])

$$x_i^j(t+1) = x_i^j(t) + \phi_{ij}(x_i^j(t) - x_k^j(t)), \;\; j = \overline{1,n}$$

Instead of conclusions