Artificial Life

4 – Swarm intelligence

Stefan Bornhofen
Vincent Gardeux
1) Complexity, Emergence & CA (sb)
2) Fractals and L-systems (sb)
3) Multi-agent systems (vg)
4) Swarm intelligence (vg)
5) Artificial evolution (vg)
6) Virtual Ecosystems & Perspectives (sb)
Inspired from natural insects:

- Ants, Bees or Termites (all *social insects*) show impressive collective problem-solving capabilities

- They develop interesting group behavior: self-organisation, robustness and flexibility => used in *artificial systems* for optimization, control or task execution.

- That introduces the idea of developing algorithms inspired by their strictly self-organised behavior.
Social insects work without supervision. Their teamwork is largely self-organized, and coordination arises from the different interactions among individual in the colony.

Although these interactions might be primitive, taken together they result in efficient solutions to difficult problems.

The collective behavior that emerges from a group of social insects has been dubbed swarm intelligence.
Swarm Intelligence is a concept that:

- describes the collective behavior of decentralized, self-organized systems, natural or artificial.
- was introduced by Gerardo Ben and Jing Wang in 1989, in the context of cellular robotic systems.
Principle of an artificial SI System:

- It’s made up by a population of mobile agents that can interact with other agents and/or with their environment locally (direct or indirect communication = stigmergy).
- Each agent follows simple rules but there is **not** global behavior control system.
- They can **collectively** carry out a distributed problem solving.
Results obtained by SI System:

- Global intelligent behavior called **Collective Intelligence** emerge from the interactions between the agents.

- The agents are not aware of this global intelligence, they are a part of the system and only follow simple rules.
Advantages of Swarm Intelligence:

- **Flexibility**: The swarm can quickly adapt to a changing environment.

- **Robustness**: Even when one or more individuals fail, the swarm can still perform its tasks.

- **Self-Organization**: The swarm needs relatively little supervision or top-down control.
Applications

Two major groups:

In Optimization:
- Ant Colony Optimization (ACO).
- Particle Swarm Optimization (PSO).
- ...

In Simulation (Social Analogy):
- Crowd Simulation
- Flocking Simulation
- Swarm Robots
- ...

Ant Colony Optimization

- It’s an heuristic proposed by Marco Dorigo in 1992 in his PhD thesis.
- Originally, the algorithm was aiming to search for an optimal path in a graph.
- It’s based on the behavior of ants seeking a path between their colony and a source of food.
- Now, ACO are used to solve a large diversity of optimization problems.
Ants in Nature

- Ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails.
- If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food.
- Ants so communicate themselves using their environment: this is called stigmergy.
Ants in Nature

- The pheromone trail **evaporates** over time, thus reducing its attractive strength.
- So, the more it takes to an ant to come back to its colony, the more the trail will be evaporated.
- Thus the pheromone density tends to remain higher on the shortest path as the ant comes back to the colony faster.
Ant Colony Optimization

1. N : Nest
2. F : Food
3. N : Nest
Artificial ants

- The idea of ACO is to mimic this behavior with artificial ants walking around the graph representing the problem to solve.
- Often used to solve Travelling Salesman Problem (TSP), Routing Vehicle, KnapSack, ...
Ant Colony Optimization

ACO Algorithm for TSP:

Initialize the graph of pheromones Gp.
While a stopping criterion is not reach
   For i from 1 to nbAnts
      Generate the path of the ith ant: it starts to an initial vertex and compute the next one. For each vertex, a probability to be selected is computed following Gp (more pheromons => more chance to be choose) and a random behavior and a particular vertex is then selected. Repeated until all vertices are processed.
      Update Gp: reinforce pheromons on the edges used depending on the found solution fitness (can be done only by the best ant)
   End For
   Evaporate pheromons on the edges by a given percentage p
End While
Ant Colony Optimization

ACO Algorithm for TSP:

1

2

3

4
Ant Colony Optimization

ACO Algorithm for TSP:
Artificial ants

- Pheromone evaporation allows avoiding the convergence to a locally optimal solution.
- If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones.
- In that case, the exploration of the solution space would be constrained.
Particle Swarm Optimization

- Created by Russel Eberhart and James Kennedy in 1995.
- Inspired by a model developed by a biologist: Craig Reynolds in the 80ies that simulate the flocking behavior of birds.
- Based on the collaboration between agents.
- Designed originally for continuous problems.
- Each agent (particle) has simple moving rules to follow and the swarm converge quickly to a local optimum.
Particle Swarm Optimization

- The swarm is typically modeled by particles in multidimensional space that have a position and a velocity.

- These particles fly through hyperspace and have two essential reasoning capabilities: their memory of their own best position and knowledge of the global or their neighborhood's best solution.
Particle Swarm Optimization

PSO Algorithm:

- Initially, a swarm of particles is generated randomly in the search space.
- At each iteration, each particle moves according to:
  - It’s current speed
  - It’s own best solution
  - The best solution found by all particles (in a limited neighborhood or in all population)
Particle Swarm Optimization

Example resolution of a function for $D = 2$. 
Particle Swarm Optimization

- A social network can be simulated in order to select the informators.

- A huge number of hybridation and variants have been made, the most popular is **Tribes**.

- Quick convergence to local optima

- Easy to implement and few parameters to tune.
Crowd Simulation

- Tim Burton's **Batman Returns** was the first movie to make use of swarm technology for rendering, realistically depicting the movements of a group of bats (BOIDS)
- **Lord of the Rings** film trilogy made use of similar technology (MASSIVE), during battle scenes

Why using Swarm Intelligence in these applications?

- Realistic movements of a simulated crowd
- Cheap and Simple to use
Flocking Simulation:

A well-known artificial life program, developed by Craig Reynolds in 1986, is named **BOIDS** and simulates the flocking behavior of birds.

Boids is an example of emergent behavior coming from the interaction of individual agents (the boids, in this case) adhering to a set of simple rules.

The rules applied in the simplest Boids world are as follows:

- **separation**: steer to avoid crowding local flockmates
- **alignment**: steer towards the average heading of local flockmates
- **cohesion**: steer to move toward the average position of local flockmates
More complex rules can be added, such as obstacle avoidance and goal seeking.

The movement of Boids can be characterized as either chaotic (splitting groups and wild behavior) or orderly.

Unexpected behaviors, such as splitting flocks and reuniting after avoiding obstacles, can be considered emergent.

The boids framework is often used in computer graphics, providing realistic-looking representations of flocks of birds and other creatures, such as schools of fish or herds of animals.
Simulation

- BOIDS Example:

```
COURSE: 07
COURSE ORGANIZER: DEMETRI TERZOPoulos

"BOIDS DEMOS"
CRAIG REYNOLDS
SILICON STUDIOS, MS 3L-980
2011 NORTH SHORELINE BLVD.
MOUNTAIN VIEW, CA  94039-7311
```
Simulation

- Application in predator/prey simulation:
Application in Fish School simulation:
Swarm Robots:
- Multiple robots working together on a single task.
- Groups of robots can perform these tasks more efficiently, and can perform them in fundamentally different ways than robots working individually.

Areas of application:
- Vacuum cleaner robots might need to share maps of areas where they've previously cleaned.
- Mars rovers might need to disperse throughout the environment to locate promising areas, while maintaining communications with each other.
- Earthquake rescue robots might infiltrate the debris and locate survivors.
Individual robots work independently, only communicating with other nearby robots is possible.

It can be either too expensive (robot vacuums need to be very cheap), too far (it takes 15 minutes for messages to get to Mars), or impossible (radio control signals cannot penetrate into earthquake rubble) to control all of the robots from a centralized location.

However, a distributed control system can let robots interact with other nearby robots, cooperating amongst themselves to accomplish their mission.
Simulation

Examples:

Dispersing

Clumping

Video Shown 4X Speed

Follow the leader
Simulation
References


http://www.swarmrobot.org/
http://www.red3d.com/cwr/boids/
http://www.swarmintelligence.org/