# Ant Colony

Overview

Pat Riddle

# Natural Inspiration

- The name *Ant Colony Optimization* was chosen to reflect its original inspiration: the foraging behavior of some ant species.
- It was inspired by the double-bridge experiment performed by Jean-Louis Deneubourg and colleagues.
- ants are able to find the shortest path to a food source by collectively exploiting pheromones they deposit on the ground while moving.
- Although ACO has grown to become a fully fledged algorithmic framework and now includes many components that are no longer related to real ants

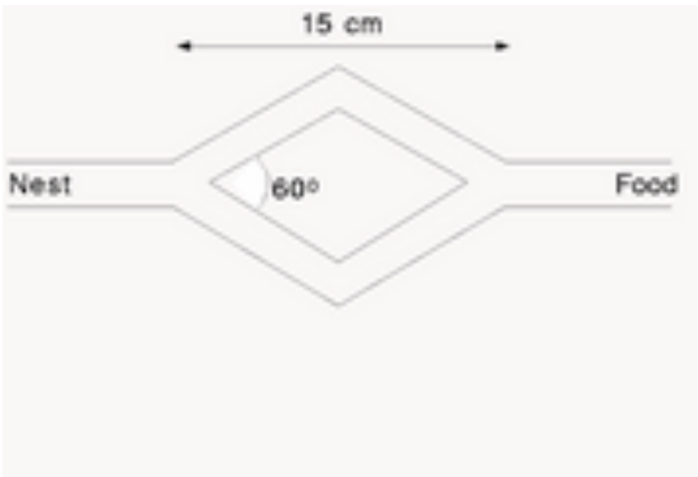
# Double Bridge Experiment

- a nest of a colony of *Argentine ants* is connected to a food source by two bridges.

- The ants can reach the food source and get back to the nest using any of the two bridges.

- The goal of the experiment is to observe the resulting behavior of the colony.

# Double Bridges




# Bridges the same Length

- if the two bridges have the same length, the ants tend to converge towards the use of one of the two bridges.

- If the experiment is repeated a number of times, it is observed that each of the two bridges is used in about 50% of the cases.

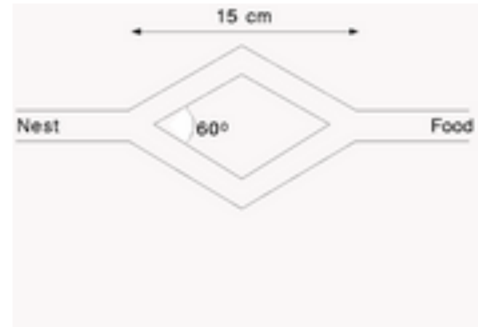
# Ant Movement Rules

- while moving, ants deposit pheromone on the ground;

- whenever they must choose which path to follow, their choice is biased by pheromone:

- the higher the pheromone concentration found on a particular path, the higher is the probability to follow that path.

# Ant Behavior




- How the ants converge towards the use of a single bridge?
  - At the start of the experiment the ants explore the surroundings of the nest.
  - When they arrive at the decision point in which they have to choose which of the two bridges to use, they choose probabilistically biased on the pheromone
  - Initially, as there is no pheromone yet, each ant chooses one of the two bridges with 50% probability
  - However, after some time, because of random fluctuations, one of the two bridges presents a higher concentration of pheromone than the other and, therefore, attracts more ants.
  - This in turn increases the pheromone level on that bridge, making it more attractive.
- It is this autocatalytic mechanism that makes the whole colony converge towards the use of the same bridge.

# Short Bridge Experiment


- If one of the bridges is significantly shorter than the other, a second mechanism plays an important role:
  - the ants that happen randomly to choose the shorter bridge are the first to reach the food source.
  - When these ants, while moving back to the nest, encounter the decision point 2,
    - they sense a higher pheromone on the shorter bridge,
    - which is then chosen with higher probability and
    - once again receives additional pheromone.

- This fact increases the probability that further ants select the short one rather than the long one.

# Ant Colony Overview

- Ant colony optimization is a population-based metaheuristic that can be used to find approximate solutions to difficult optimization problems.

- In ant colony optimization (ACO), a set of software agents called "artificial ants" search for good solutions to a given optimization problem transformed into the problem of finding the minimum cost path on a weighted graph.

- The artificial ants incrementally build solutions by moving on the graph.

- The solution construction process is stochastic and is biased by a pheromone model
  - a set of parameters associated with graph components (either nodes or edges) the values of which are modified at runtime by the ants.

# ACO Successes

- ACO has been applied successfully to many classical combinatorial optimization problems, as well as to discrete optimization problems that have stochastic and/or dynamic components.

- Examples: routing in communication networks and stochastic versions of well-known combinatorial optimization problems, such as the probabilistic traveling salesman problem.

- ACO has been extended so that it can be used to solve continuous and mixed-variable optimization problems (Socha and Dorigo 2004).

# Evaporation

- In the real world, 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 traveling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food.
- Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength.
- The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate.

# Avoiding Local Optima

- A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate.
- Pheromone evaporation has also the advantage of 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.

# Positive Feedback

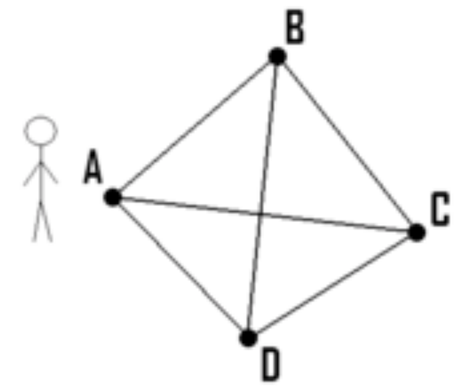
- Thus, when one ant finds a good (i.e. short) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads all the ants to follow a single path.

- The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph representing the problem to solve.

# ACO Advamtage

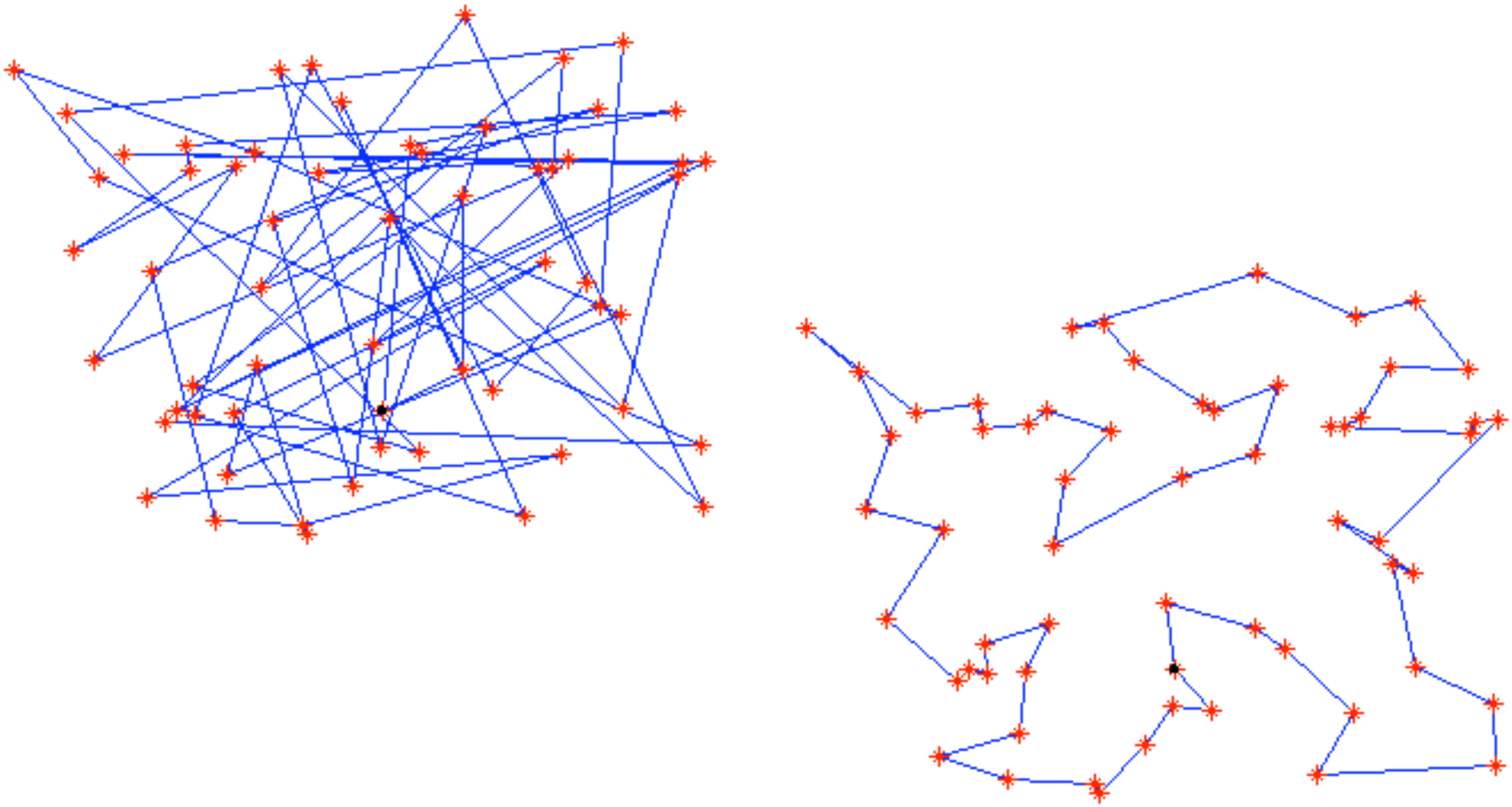
- Ant colony optimization algorithms have been used to produce near-optimal solutions to the traveling salesman problem.
- They have an advantage over simulated annealing and genetic algorithm approaches when the graph may change dynamically;
  - the ant colony algorithm can be run continuously and adapt to changes in real time.
- This is of interest in network routing and urban transportation systems.

# TSP example




- In the traveling salesman problem (TSP) a set of locations (cities) and the distances between them are given.
- The problem consists of finding a closed tour of minimal length that visits each city once and only once.
- To apply ACO to the TSP, we consider the graph defined by associating the set of cities with the set of vertices of the graph.
- This graph is called the *construction graph*.

# Larger TSP example



# Construction Graph

- In TSP it is possible to move from any city to any other city
- The construction graph is fully connected and the number of vertices is equal to the number of cities.
- The pheromone values and heuristic values are associated with the edges of the graph.
- Pheromone values are modified at runtime and represent the cumulated experience of the ant colony
- The heuristic values are problem dependent values that, in the case of the TSP, are the inverse of lengths of the edges.

# The Solution

- The ants construct the solutions as follows.
  - Each ant starts from a randomly selected city (vertex of the construction graph).
  - Then, at each construction step it moves along the edges of the graph.
  - Each ant keeps a memory of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited.
  - An ant has constructed a solution once it has visited all the vertices of the graph.

# Probabilistic Rule

- At each construction step, an ant probabilistically chooses the edge to follow from those that lead to yet unvisited vertices.

- The probabilistic rule is biased by pheromone values and heuristic information:
  - the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will choose that particular edge.

# Pheromone Update

- Once all the ants have completed their tour, the pheromone on the edges is updated.
- Each of the pheromone values is initially decreased by a certain percentage.
- Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs (there is one solution per ant).
- This procedure is repeatedly applied until a termination criterion is satisfied.

# The Ant Moves
## (…or was it “the turtle moves?”)


- The ants move from vertex to vertex along the edges of the construction graph exploiting information provided by the pheromone values and incrementally building a solution.

- the ants deposit a certain amount of pheromone on the edges that they traverse.

- The amount $\Delta\tau$ of pheromone deposited may depend on the quality of the solution found.

- Subsequent ants utilize the pheromone information as a guide towards more promising regions of the search space.

# Applications of ACO

- The initial applications of ACO were in the domain of NP-hard combinatorial optimization problems.

- An overview of these applications (Dorigo & Stützle 2004).

- Another application that was considered early in the history of ACO is routing in telecommunication networks.

- An example of ACO algorithm in this domain is AntNet (Di Caro & Dorigo 1998).

# Current ACO Trends

- Current research in ACO algorithms is devoted both to the development of theoretical foundations and to the application of the metaheuristic to new challenging problems.
- Concerning theoretical foundations:
  - The development of a theoretical foundation was started by Gutjahr, who was the first to prove convergence in probability of an ACO algorithm (Gutjahr 2000).
  - An overview of theoretical results available for ACO can be found in (Dorigo & Blum 2005).
- Concerning applications:
  - the use of ACO for the solution of dynamic, multiobjective, stochastic, continuous and mixed-variable optimization problems is a hot topic,
  - the creation of parallel implementations capable of taking advantage of the new available parallel hardware.
- Many papers reporting on current research can be found in the proceedings of the ANTS conference or in the Swarm Intelligence journal.

# Telecommunication Applications

- The use of Swarm Intelligence in Telecommunication Networks has also been researched, in the form of Ant Based Routing.

- This was pioneered separately by Dorigo et al and Hewlett Packard in the mid-1990s, with a number of variations since.

- Basically this uses a probabilistic routing table rewarding/ reinforcing the route successfully traversed by each "ant" (a small control packet) which flood the network.

- As the system behaves stochastically and is therefore lacking repeatability, there are large hurdles to commercial deployment.

# Particle Swarm Optimization

Overview

Patricia J Riddle

# What is PSO?

- **Particle swarm optimization (PSO)** is a swarm intelligence based algorithm to find a solution to an optimization problem in a search space, or model and predict social behavior in the presence of objectives.

- Particle swarm optimization is a stochastic, population-based evolutionary computer algorithm for problem solving.

- It is a kind of swarm intelligence that is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications.

- The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart.

# PSO is like GAs

- Particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling.

- The system is initialized with a population of random solutions and searches for optima by updating generations. (JUST LIKE GAs)

# PSO is not like GAs

- PSO has no evolution operators such as crossover and mutation.
- The potential solutions, called particles, fly through the problem space by following the current optimum particles.
- Compared to GAs, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust.
- PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

# Social Learning

- Social influence and social learning enable a person to maintain cognitive consistency.

- People solve problems by talking with other people about them, and as they interact their beliefs, attitudes, and behaviors change; the changes could typically be depicted as the individuals moving toward one another in a sociocognitive space.

# Particle Swarm

- The particle swarm simulates this kind of social optimization.
- A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function.
- A communication structure or social network is also defined, assigning neighbors for each individual to interact with.
- Then a population of individuals defined as random guesses at the problem solutions is initialized – **candidate solutions**.
- They are also known as the particles, hence the name particle swarm.

# The Algorithm

- An iterative process to improve these candidate solutions is set in motion.
- The **particles iteratively evaluate the fitness** of the candidate solutions and **remember the location where they had their best success**.
- The individual's best solution is called the **particle best or the local best**.
- Each particle makes this information available to their neighbors.
- They are also able to see where their neighbors have had success.
- Movements through the search space are guided by these successes, with the population converging, by the end of a trial, on a problem solution

# Particles

- The swarm is typically modelled by particles in multidimensional space that have a **position** and a **velocity**.
- These particles fly through hyperspace (i.e., $\Re^n$) and have two essential reasoning capabilities:
  - their memory of their **own best position** and
  - knowledge of the **global or their neighborhood's best**.
- In a minimization optimization problem, "best" simply meaning the position with the smallest objective value.
- Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions.

# Particle Information

- So a particle has the following information to make a suitable change in its position and velocity:
  - A global best that is known to all and immediately updated when a new best position is found by any particle in the swarm.
  - Neighborhood best that the particle obtains by communicating with a subset of the swarm.
  - The local best, which is the best solution that the particle has seen.

# Particle Position

- The particle position and velocity update equations in the simplest form that govern the PSO are given by
- $$\begin{aligned} v_{i,j} \leftarrow\ & c_0 v_{i\,j} + \\ & c_1 r_1 (globalbest_j - x_{i,j}) + \\ & c_2 r_2 (localbest_{i,j} - x_{i,j}) + \\ & c_3 r_3 (neighborhoodbest_j - x_{i,j}) \end{aligned}$$

$$x_{i,j} \leftarrow x_{i,j} + v_{i,j}$$

- *i* is the particle and *j* is the dimension
- r1,r2,r3 are random numbers
- c1 c2 c3 are learning factors

# Convergence

- As the swarm iterates, the fitness of the global best solution improves (decreases for minimization problem).
- It could happen that all particles being influenced by the global best eventually approach the global best, and from there on the fitness never improves despite however many runs the PSO is iterated thereafter.
- The particles also move about in the search space in close proximity to the global best and not exploring the rest of search space – convergence

# The Power

- A single particle by itself is unable to accomplish anything.

- The power is in interactive collaboration.

# Evolutionary Techniques

- Most of evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop. Otherwise go back to 2.

# Similarities

- So PSO shares many common points with GA.
  - Both algorithms start with a group of a randomly generated population
  - Both have fitness values to evaluate the population.
  - Both update the population and search for the optimum with random techniques.
  - Both systems do not guarantee success.

# Differences

- PSO does not have genetic operators like crossover and mutation.
  - Particles update themselves with the internal velocity.
  - They also have memory, which is important to the algorithm.

- Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different.

# One Way Information Sharing

- In GAs, chromosomes share information with each other.
- So the whole population moves like a one group towards an optimal area.
- In PSO, only gBest (or lBest) gives out the information to others.
  - It is a one -way information sharing mechanism.
  - The evolution only looks at the best solution.
- Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases.