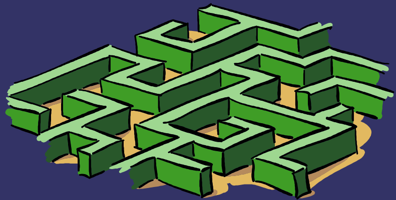
Swarm Intelligence and Ant Colony Optimization

Background, Methods, and Applications

Lucas deHart

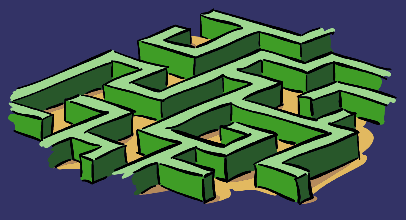
What is Swarm Intelligence?

- Computer simulation of biological processes
- Takes inspiration from animals such as bees and ants



What is Swarm Intelligence?

- Best described as “Optimization without knowledge”
- Each agent follows simple rules – no higher understanding of the problem
- When many agents act together, a collective intelligence emerges

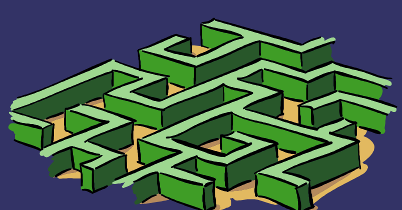


What is Swarm Intelligence?

- Foraging behavior among insects
- Scouts are sent out from a hive to find food
- Once found, scouts return to tell the hive, and more members of the hive follow the same path
- Efficient Searching?

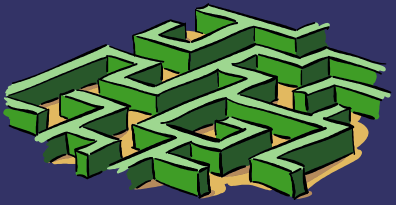
Properties of Swarm Intelligence

- Large number of “individuals” forms a “population” of possible solutions
- Individuals are fairly homogeneous – little variation between types
- Individuals operate autonomously, but interact with each other and their environment
- The group behavior self-organizes



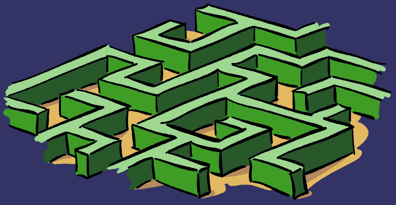
Properties Continued

- Scalability – Can increase scale without reprogramming interactions.
    - Number of interactions scales slowly with population size
- Easy to parallelize – Individual actions depend only on close neighborhood.
- Fault tolerance – Faulty individuals have a small impact and can be easily replaced.



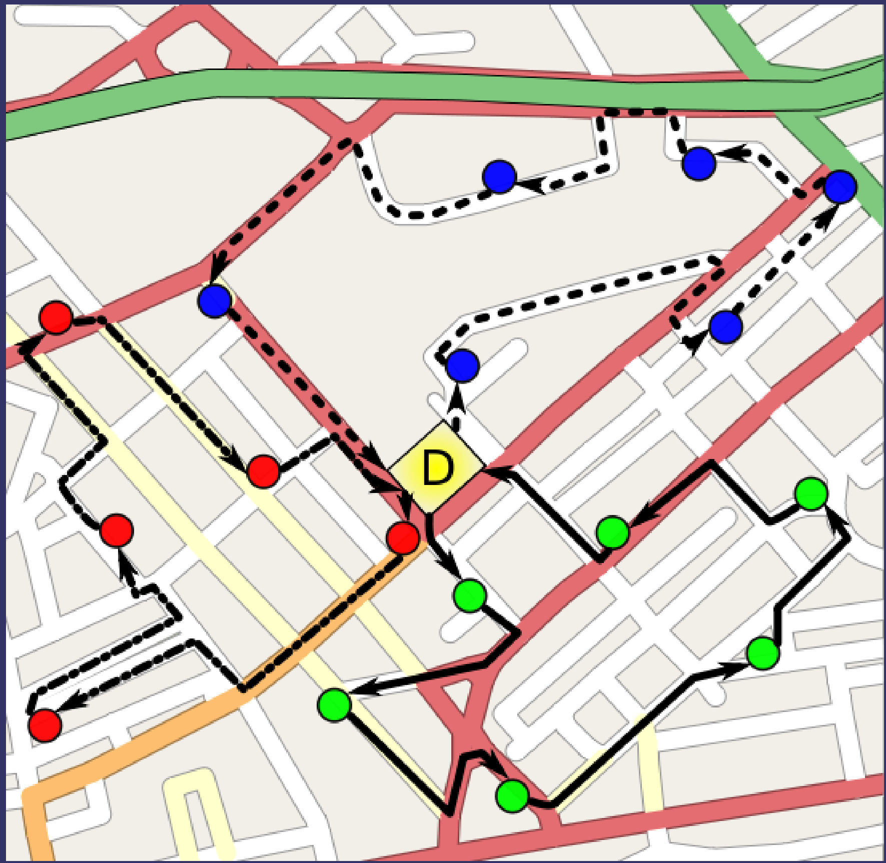
The Swarm Intelligence Family

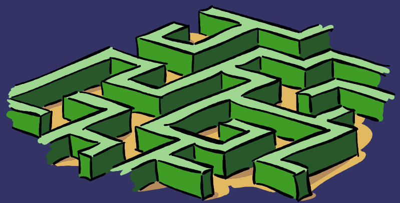
- Swarm intelligence is a computing approach, rather than a particular algorithm
- It can be further divided and categorized. Examples of algorithms within swarm intelligence include:
  - Ant Colony Optimization
  - Particle Swarm Optimization
  - The Bee Algorithm



Applications of Swarm Intelligence Algorithms

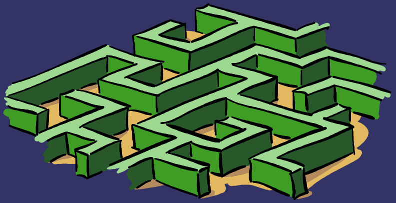
- Modeling real life swarming behavior
- Swarm Robotics
- Optimization Problems
  - Discrete and Continuous
    - Vehicle Routing, for example





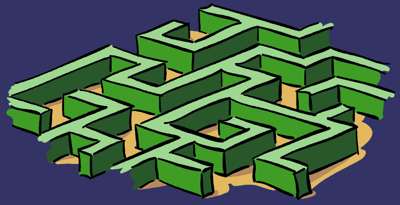
Ant Colony Optimization Algorithms

- Based on the foraging practices of ant hives
- Not a specific Algorithm, but a related family
- Ants leave pheromone trails behind them
- Pheromone trails affect the likelihood that a subsequent ant will follow the same path
- Pheromone trails evaporate, helps prevent convergence to local minima or maxima



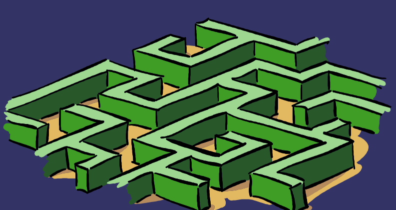
ACO: Basic Algorithm

- ACO Solves Combinatorial Optimization Problems
  - Finite number of decision variables → Search Space (S)
  - Constraints among variables (Omega)
  - Function f:S → R
  - Solutions are any set of variables in the search space S that satisfy the constraints Omega
  - Lets look at a specific problem



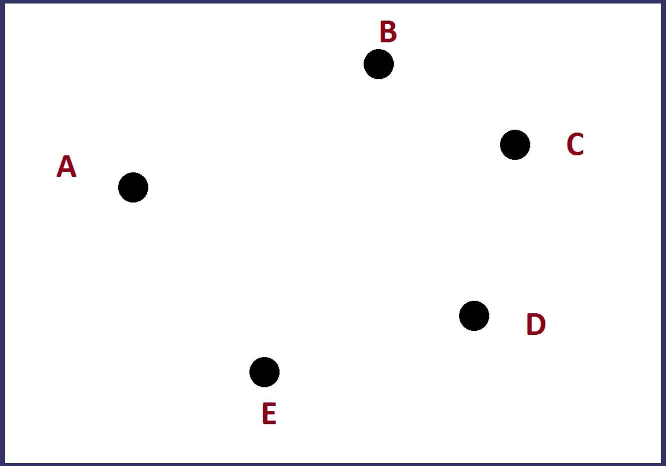
Traveling Salesman Problem

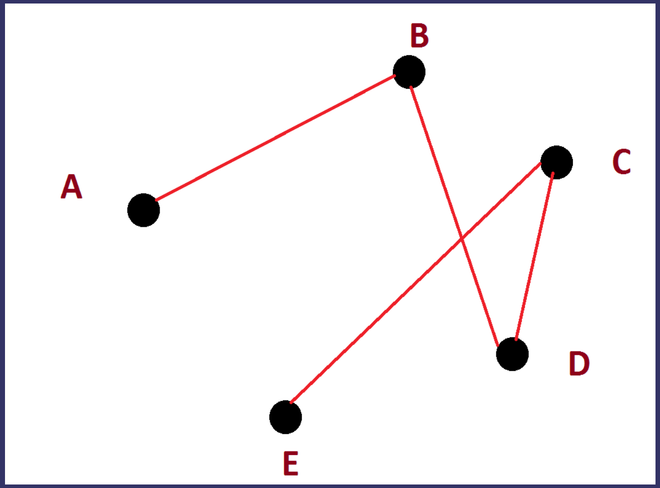
- A salesman wants to visit each city in an area while ensuring the following things are true:
  - The salesman visits every city
  - The salesman visits each city only once
- Optimize the solution by finding the shortest path for the salesman to take while still obeying the constraints

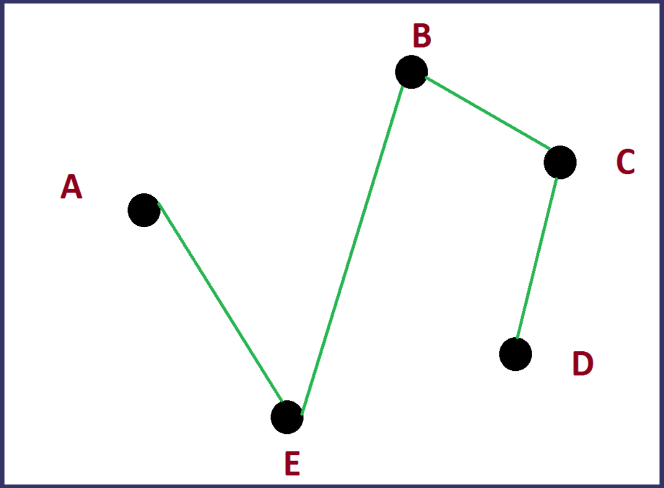


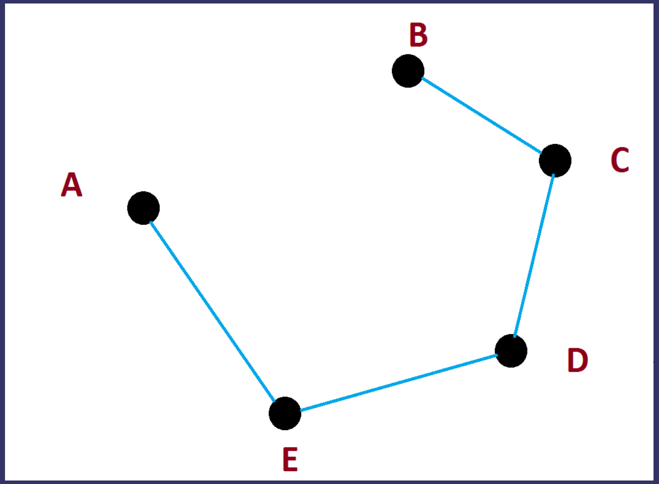
Traveling Salesman Problem

- Consider a simple map of five cities
- Spatially distributed on a plane
- Considering city A as start point
- Three possible solutions to TSP



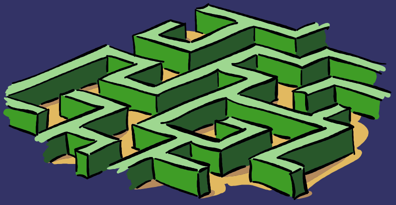






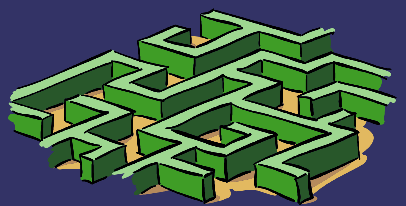
Applying ACO to TSP

- We will apply an Ant Colony Optimization Algorithm to the TSP
  - The original ACO Algorithm was developed with TSP in mind
  - TSP is a simple problem to describe, but with a large number of nodes, is difficult to solve well



Step 1: Initialization

- Build the search space S and initialize pheromone levels
  - For each city define a spatially distributed vertex
  - Define edges as connections between vertices
  - For every edge, set pheromone level to t0
- Distribute m ants among the vertices
  - Add the start locations to the solution memory of each ant



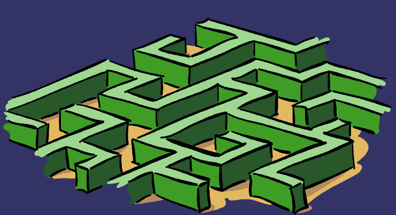
Step 2: Construct Ant Solutions

- Iterative solution construction steps occur for each ant
- At each Construction Step, ants add feasible partial solutions to their memory
  - For TSP, partial solutions are steps from one city to another
  - Choice of partial solution determined probabilistically
- Once chosen, deposit pheromone on the edge traveled
- Iterate n-1 times (until complete solutions are formed)

Step 2.5: How do they choose?

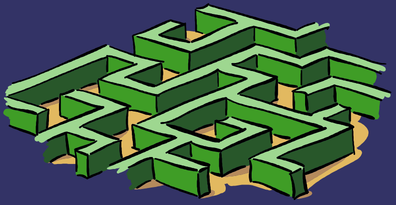
- The Ant-Routing Table (Ai) holds probabilities that an ant will make a particular move
- The pheromone values for a move are given as t
- The nij is the inverse of the distance between the two cities

$$a_{ij} = \frac{[t_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in A_i}[t_{il}]^{\alpha}[\eta_{il}]^{\beta}} \forall_{j} \in N_{i}$$



Step 3 (optional): Apply Local Search

- After obtaining full solutions, local search may be applied in the area of those solutions to improve them further between global iterations
- For a very simple TSP, this is unnecessary
- For more complex problems, Local Search Algorithms can increase the performance of ACO

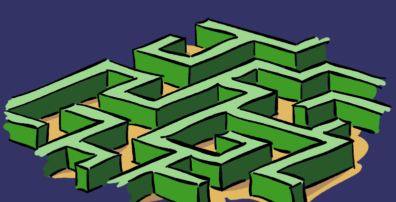


Step 4: Global Pheromone Update

- After each global iteration (once a set of complete solutions is found), the pheromone levels of each edge are updated.

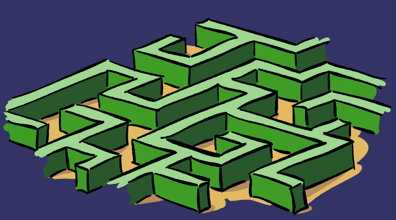
$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{s \in S_{\mathrm{upd}} | c_i^j \in s} g(s)$$

- g(s) is the evaluation function, or how good the particular solution is
- Evaporation helps prevent fast convergence to a local minimum, instead driving the algorithm to search new areas of the search space



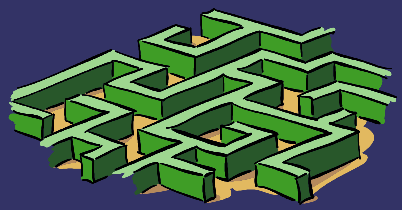
Step 5: End

- The Global Iteration is performed a set number of times
- After the conclusion of the iterations, we have a large number of edges with pheromone values
- Because pheromone is distributed more heavily on the best sections of a solution, we can pick the final solution based on the pheromone levels
- Can build a final solution from good solution parts that is better than any individual solution
- No single ant has to traverse the “best” solution



Further Applications

- We have shown how ACO works on a static problem
- However, ACO also works for dynamic problems
  - Dynamic Problems involve changing search spaces
- Adaptation is fairly simple, just let the algorithm run continuously
- Has been used to solve urban traffic problems and telephone network routing



References

1. http://www.scholarpedia.org/article/Swarm_intelligence
2. http://www.scholarpedia.org/article/Ant_colony_optimization
3. Ant Colony Optimization, Saad Ghaleb Yaseen and Nada M A.AL-Slamy, 2008
4. Ant Colony Optimization, Overview and Recent Advances, Marco Dorigo and Thomas Stutzle, 2009
5. http://staff.washington.edu/paymana/swarm/krink_01.pdf

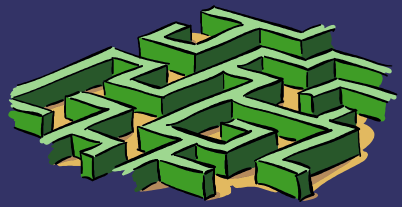


Image References

1.https://commons.wikimedia.org/wiki/File:Bees_Collecting_Pollen_2004-08-14.jpg

2. https://commons.wikimedia.org/wiki/File:Weaver_ant_carrying_food.jpg

3. https://upload.wikimedia.org/wikipedia/commons/e/e0/Vehicle_Routing_Problem_Example.svg