Ant Colony Optimization

EDWIN WONG
PHILLIP SUMMERS
ROSALYN KU
PATRICK XIE
PIC 10C SPRING 2011

Swarm Intelligence

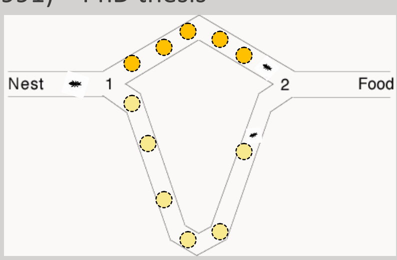
- Swarms
  - Swarm of bees
  - Ant colony as swarm of ants
  - Flock of birds as swarm of birds
  - Traffic as swarm of cars
  - Immune system as swarm of cells and molecules
  - ...
- Swarm Intelligence/Agent Based Modeling
  - Model complex behavior using simple agents

Swarm Intelligence

- Digital Crumbs a la Hansel and Gretel
  - Idea: **stigmergy** is a mechanism of communication by modifying the environment
  - Example
    - Take some dirt in your mouth
    - Moisten it with pheromones
    - Walk in the direction of the strongest pheromone concentration
    - Drop what you are carrying where the smell is the strongest
  - Ant Colony Optimization uses artificial stigmergy

Swarm Intelligence

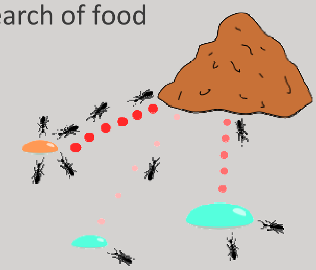
- Ant Colony Optimization
  - Marco Dorigo (1991) – PhD thesis



  - Technique for solving problems which can be expressed as finding good paths through graphs
  - Each ant tries to find a route between its nest and a food source

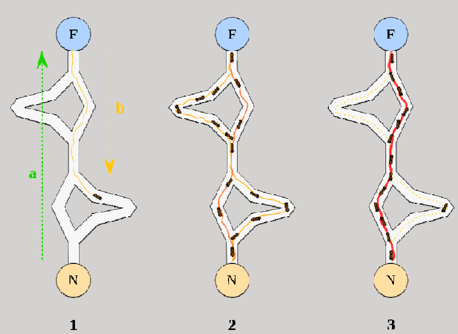
Swarm Intelligence

- The behavior of each ant in nature
  - Wander randomly at first, laying down a pheromone trail
  - If food is found, return to the nest laying down a pheromone trail
  - If pheromone is found, with some increased probability follow the pheromone trail
  - Once back at the nest, go out again in search of food
- However, pheromones evaporate over time, such that unless they are reinforced by more ants, the pheromones will disappear.



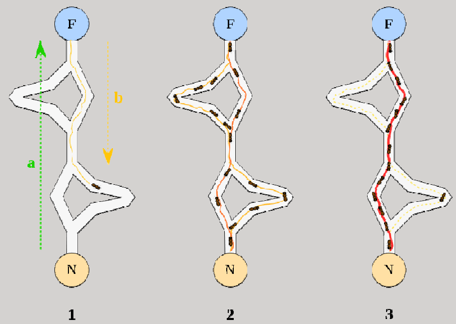
Ant Colony Optimization

1. The first ant wanders randomly until it finds the food source (F), then it returns to the nest (N), laying a pheromone trail



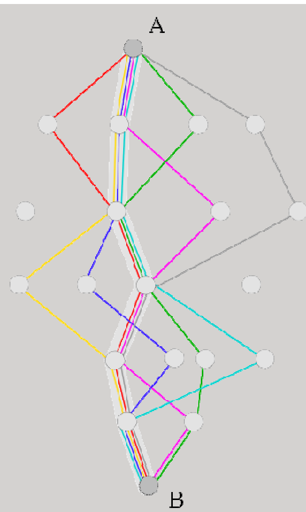
Ant Colony Optimization

2. Other ants follow one of the paths at random, also laying pheromone trails. Since the ants on the shortest path lay pheromone trails faster, this path gets reinforced with more pheromone, making it more appealing to future ants.
3. The ants become increasingly likely to follow the shortest path since it is constantly reinforced with a larger amount of pheromones. The pheromone trails of the longer paths evaporate.



Ant Colony Optimization

- Paradigm for optimization problems that can be expressed as finding short paths in a graph
- Goal
  - To design technical systems for optimization, *and*
  - NOT to design an accurate model of nature



Ant Colony Optimization

Nature	Computer Science
Natural habitat	Graph (nodes and edges)
Nest and food	Nodes in the graph: start and destination
Ants	Agents, our artificial ants
Visibility	The reciprocal of distance, η
Pheromones	Artificial pheromones ,τ
Foraging behavior	Random walk through graph (guided by pheromones)

Ant Colony Optimization

- Scheme:
  - Construct ant solutions
  - Define attractiveness τ, based on experience from previous solutions
  - Define specific visibility function, η, for a given problem (e.g. distance)
- Ant walk
  - Initialize ants and nodes (states)
  - Choose next edge probabilistically according to the attractiveness and visibility
  - $$Prob(choose\ available\ edge\ e) = \frac{\tau(e) \bullet \eta(e)}{\sum_{available\ edges\ e'} \tau(e') \bullet \eta(e')}$$
  - Each ant maintains a tabu list of infeasible transitions for that iteration
  - Update attractiveness of an edge according to the number of ants that pass through

Ant Colony Optimization

- Pheromone update

$$\tau(e) := \begin{cases} (1-\rho)\cdot\tau(e), & \textit{if edge is not traversed} \\ (1-\rho)\cdot\tau(e) + \textit{new pheromone}, & \textit{if edge is traversed} \end{cases}$$

- Parameter $0 \leq \rho \leq 1$ is called **evaporation rate**
- Pheromones = **long-term memory** of an ant colony
  - ρ small → low evaporation → slow adaptation
  - ρ large → high evaporation → fast adaptation
- Note: rules are probabilistic, so mistakes can be made!
- "new pheromone" or Δτ usually contains the base attractiveness constant Q and a factor that you want to optimize
  - (e.g.) Q/length of tour

General Ant Colony Pseudo Code

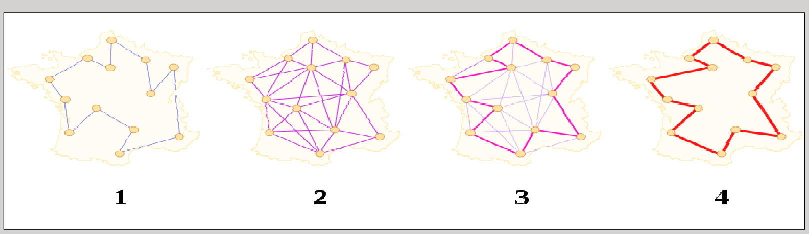
Initialize the base attractiveness, τ, and visibility, η, for each edge;
for i < IterationMax do:
    for each ant do:
        choose probabilistically (based on previous equation) the next state to move into;
        add that move to the tabu list for each ant;
        repeat until each ant completed a solution;
    end;
    for each ant that completed a solution do:
        update attractiveness τ for each edge that the ant traversed;
    end;
    if (local best solution better than global solution)
        save local best solution as global solution;
    end;
end;

Heuristic Information

- Heuristics refers to experience-based techniques for problem solving, learning, and discovery
- Prime example: trial and error
- In computer science, **metaheuristic** is a computational method that optimizes a problem by iteratively trying to improve a candidate solution
  - Example: black box, cracking a combination lock, planning a route from Miami to Dallas
- Metaheuristics allows us to find the best solution over a discrete search-space

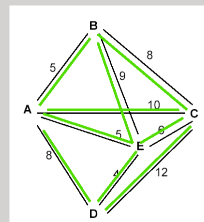
Traveling Salesman Problem

- Traveling Salesman Problem (TSP)
  - In the Traveling Salesman Problem (TSP) a salesman visits *n* cities once.
  - Problem: What is the shortest possible route?



Solutions?

- Brute Force Method:
  - Create permutations for all N cities within the TSP
  - Iteratively check all distances
  - Can you guys figure out the Big O notation for such a problem?
- Greedy Algorithm:
  - Searches for locally optimal solutions



ACO and the Traveling Salesman Problem

- An artificial ant *k* has a memory of the cities that it has already visited, M_k *or tabu*
- Add **heuristic information** to ant walk: τ(e) describes the attractiveness of an edge
- $\eta(e) = 1/d$
  - inverse distance (visibility) between cities
- An ant *k* in city *i* chooses the next city according to

$$Prob(choose\ edge\ e = (i,j)) = \begin{cases} \dfrac{[\tau(e)]^{\alpha} \cdot [\eta(e)]^{\beta}}{\sum_{e'=(i,j')}[\tau(e')]^{\alpha} \cdot [\eta(e')]^{\beta}}, & if\ j \notin M_k \\ 0, & otherwise \end{cases}$$

ACO and the Traveling Salesman Problem

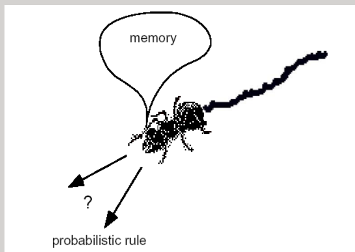
$$Prob\big(choose\ edge\ e = (i,j)\big) = \begin{cases} \dfrac{[\tau(e)]^{\alpha} \cdot [\eta(e)]^{\beta}}{\sum_{e'=(i,j')}[\tau(e')]^{\alpha} \cdot [\eta(e')]^{\beta}}, & if\ j \notin M_k \\ 0, & otherwise \end{cases}$$

- e' is an edge that the ant hasn't visited
- α and β balance impact of pheromone vs. visibility (both commonly fixed at 1)
- favors edges which are shorter and have more pheromone
- τ is the amount of pheromone on the edge (i,j)
  - $\tau = (1-\rho) * \tau + \Delta\tau^k$
  - $\Delta\tau^k = Q/L_k$, Q is constant, L_k is the length of tour of ant k

Ant System Algorithm for TSP

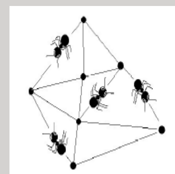
Pseudocode:

```
initialize all edges to (small) initial pheromone level τ0;
place each ant on a randomly chosen city;
for each iteration do:
    do while each ant has not completed its tour:
            for each ant do:
                        move ant to next city by the probability function
            end;
    end;
    for each ant with a complete tour do:
            evaporate pheromones;
            apply pheromone update;
            if (ant k's tour is shorter than the global solution)
                      update global solution to ant k's tour
    end;
end;
```



Benefits of Ant Colony Optimization

- Can solve certain NP-Hard problems in Polynomial time
- Directed-Random Search
  - Allows a balance between using previous knowledge and exploring new solutions
- Positive feedback for good solutions/Negative feedback for bad solutions
- Approximately convergent
- Optimal if not absolutely correct solutions
- In certain examples of ACO, no one “ant” is required to actually complete an accurate solution



Some Observed Problems

- Problem specific
  - Limited to problems that can be simulated by graphs and optimized
  - Coding difficulties for different problems
- Ineffective utilization of previously acquired information, specifically the global solution
- Depending on the design of the algorithm, it can converge towards a (less optimal) solution.

Improvements

- We might like to add factors to minimize the time it takes to reach an acceptable solution.
- Use the elements of previous solutions
  - This allows for faster convergence
  - As we construct more and more solutions, there is more information available about the probable "right" choices to make
- The decision making process might weigh exploration vs. heuristic value

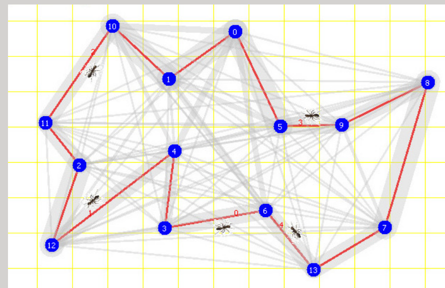
Versions of Ant Colony

- Ant System: what we just went over
- Ant Colony System:
  - Pseudo-random proportion rule:
    - at each decision point for an ant, it has a probability $(1-q_0)$ of using the same probability function as in the Ant System or q_0 of picking the best next node based on previous solutions

$$s = \begin{cases} \arg \max\limits_{(ij) \notin tabu_k} \left\{ \tau_{ij}{}^{\alpha} \cdot \eta_{ij}{}^{\beta} \right\} & \text{if } q \le q_0 \quad \text{(exploitation)} \\ \text{AS rule} & \text{otherwise} \quad \text{(exploration)} \end{cases}$$

Versions of Ant Colony

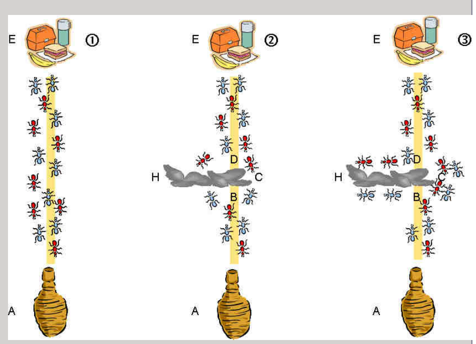
- Global Trail Update: only the best solution since the start of the computation will globally update its pheromones
- Local Trail Update: all ants consume/decrease pheromones along the path that they travel



- Elitist Ant System:
  - Both the global solution and each ant update their edges with pheromones on each iteration

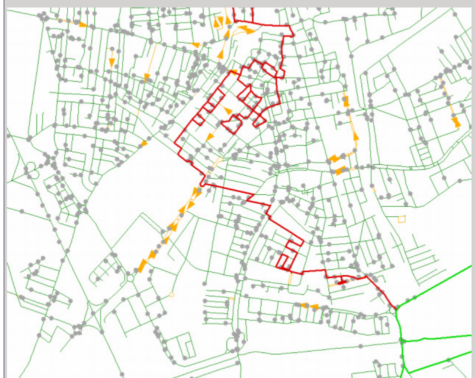
Applications

- Applications
  - Routing problems
  - Urban transportation systems
  - Facility placement
  - Scheduling problems

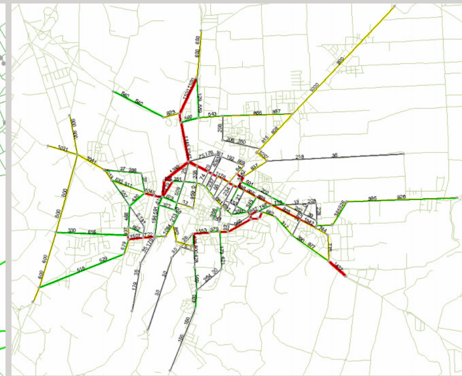


- How can we modify the algorithm?
  - Vary the importance of pheromone
  - Play around with evaporation rate
  - Add time constraint
  - Add obstacles

Applications



Urban solid waste collection



Traffic flow optimization

To sum it up:

- General paradigm for optimization problems
- Inspiration from nature, but with smarter agents
- Paths found by ant represent solutions for the problem
- Choice of path influenced by previous experience
- Pheromones as model of collective memory of a swarm
- Tunable parameters that affect performance

To see a creative implementation of Ant Colony Optimization, check out Forrest O.'s design:

http://www.openprocessing.org/visuals/?visualID=15109

References

- Dorigo M, Stützle T. Ant Colony Optimization. MIT Press; 2004
- Vittorio Maniezzo, Luca Maria Gambarde, Fabio de Luigi. http://www.cs.unibo.it/bison/publications/ACO.pdf
- Monash University CSE 460 lecture notes http://www.csse.monash.edu.au/~berndm/CSE460/Lectures/cse460-9.pdf
- "Ant colonies for the traveling salesman problem" http://www.idsia.ch/~luca/acs-bio97.pdf