Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

MSE Algorithms

L05: Ant Colony Optimization, Work on Santas Challenge

Samuel Beer

Course Overview

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

1. Introduction: Basic problems and algorithms
2. Constructive methods: Random building, Greedy
3. Local searches
4. Randomized methods
5. Threshold accepting, Simulated Annealing
6. Decomposition methods: Large neighborhood search
7. Learning methods for solution improvement: Tabu search
8. Application: Santa's Challenge
9. **Learning methods for solution building: Artificial ant systems**
10. Methods with a population of solutions: Genetic algorithms
11. Work on Santa's Challenge
12. Final Lecture

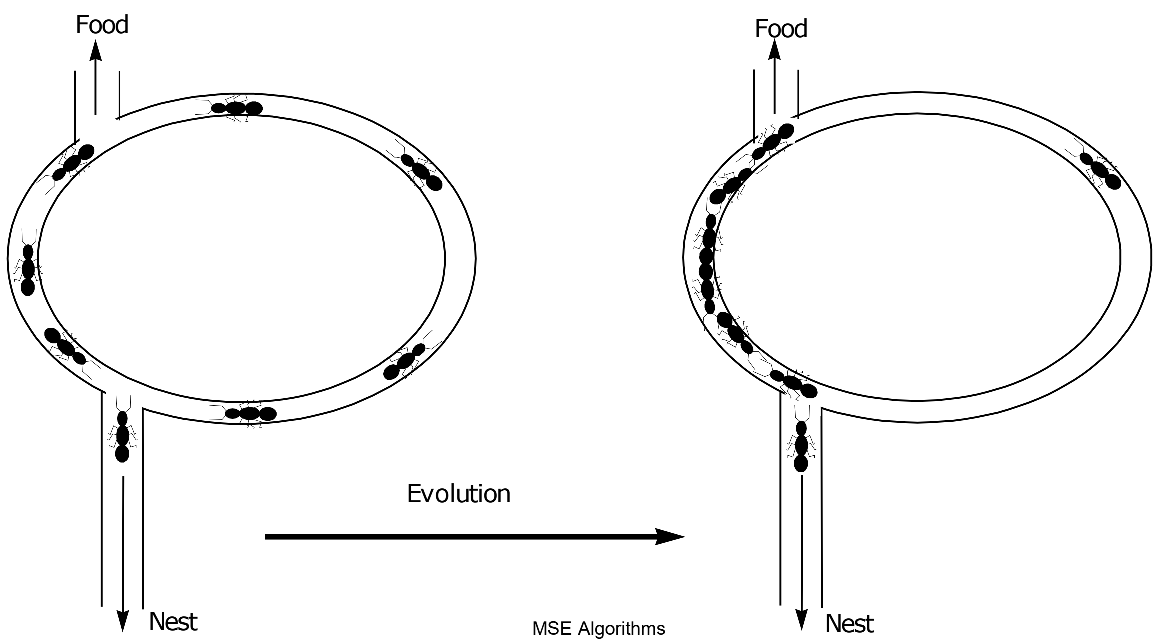
Ant Colony Optimization (ACO)

Ant Systems are used to **CONSTRUCT** solutions

Ants can find Shortest Paths

Zürcher Hochschule
für Angewandte Wissenschaften
zh
aw
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

- Ants leave a "pheromone trail" on their way from the nest to a target (e.g. food)
- The more pheromone on a segment, the more likely a next ant is to use this segment
- Experiments have shown that ants can find shortest path between source and target (2-bridge-experiment)
- Ant Colony Optimization simulates behaviour of ants to solve combinatorial problems



Meta-Heuristic: Ant Colony Optimization (ACO)

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Set parameters

Initialize pheromone trails

Do

Let artificial Ants construct solutions

Apply Local Search (optional)

Update Pheromones

Keep best solution so far

While termination condition not met

ACO is a Constructive Method, i.e. it generates new solutions

Components of an ACO System

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Memory

- Associate a value τ_e (pheromone trail) to each element *e* constituting a solution
- τ_e depends on the number of times *e* has been used ***and*** on the quality of solutions using *e*

Construct Ant Solutions (building phase)

- Let *m* artificial ants build solutions in parallel, starting from empty solutions
- For each partial solution, ants choose a feasible next element *e* with a probability proportional to $\tau_e^{\alpha} \cdot c(s,e)^{\beta}$, where *c*(*s*, *e*) is an incremental cost function that measures the quality of adding element *e* to partial solution *s*
- Parameters $\alpha \geq 0$ and $\beta \leq 0$ balance the importance of learning versus greedy choice

Update Pheromones

- **Evaporate:** Multiply all τ_e by $1 - \rho$, where $0 \leq \rho \leq 1$ is a parameter to simulate pheromone evaporation
- **Reinforcement:** For each element *e* belonging to a solution *s* at a given iteration, add Q / f(s) to τ_e, where Q is a parameter and *f(s)* the cost of solution *s*. This simulates the leaving of pheromone marks on trails by ants

Ant System for TSP

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Data
Distance matrix $D = (d_{ij})$ between cities
Trace matrix $\mathrm{T} = (\tau_{ij})$
Parameters α, β, ρ, τ_0, Q, m, *max_iter*

Initialisation
$\tau_{ij} = \tau_0$ **for all** i, j

Repeat for *max_iter* iterations
$R = (r_{ij}) = 0$ // Edge reinforcement
For each $k = 1, \ldots, m$ // Ant k build a new solution
$L = 0$ // Tour length
Choose a city i at random // Current city : i
Repeat, while all cities have not been visited
Choose a city j not yet visited with probability proportional to $\tau_{ij}^{\alpha} \cdot d_{ij}^{\beta}$
$L = L + d_{ij}$
$i = j$

Note: $\beta \leq 0$

For all arc (i, j) of the tour just built, set
$r_{ij} = r_{ij} + Q/L$

The shorter a tour of ant k, the higher the reinforcement value r_{ij} for the edges on this tour.

$\mathrm{T} = (1-\rho)\mathrm{T} + R$ // Update pheromone trail after having built m solutions

Return The best tour found

Question

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

1. What is the effect if we set $\alpha = 0$.

2. Same for $\beta = 0$.

Ant Algorithm 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

Source: Marco Dorigo et al, Ant Colony Optimization, 2006

Resources:

- *Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique* by Marco Dorigo, Mauro Birattari, and Thomas Stützle, 2006.

- https://www.researchgate.net/publication/230660829_Ant_Colony_Optimization_Artificial_Ants_as_a_Computational_Intelligence_Technique

META-HEURISTIC: MaxMin Ant System (MMAS)

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

MMAS is one of the most successful ACO variants

Changes in comparison to Classical ACO:

- Only the best ant is used to update the pheromone trails
- Maintains lower and upper bounds τ_{min} and τ_{max} for pheromone values τ_e. Updates to τ_e cannot exceed or fall below these boundaries.
- Pheromone is updated only for elements of the "best" solution found. "Best" solution could be smallest of solutions from current iteration, or "best-so-far", the best solution ever found.
- Update depends on the cost of the best solution (e.g. proportional to 1/f(s))

MMAS for TSP with Nearest Neighbors

Zürcher Hochschule für Angewandte Wissenschaften

zh aw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

```
Standard parameter settings
 α = 1, β = -2, ρ = 0.98, Q = 1, b = 20,  m = problem size
 τ_min = …, τ_max = …, max_iter = …                      // Problem dependent
Initialisation
   τ_ij = τ_max for all arc (i, j)

Repeat for max_iter iterations

   For each ant k = 1, …, m
      s_k = ∅                                           // Solution built by ant k
      Choose a city i at random                         // Current city : i
      Repeat, while all cities have not been visited
         If there is a city j not yet visited among b nearest neighbours of i
            Choose a city j among them with probability proportional to  τ_ij^α · d_ij^β

         Else Choose a city j not yet visited with maximum  τ_ij^α · d_ij^β
         s_k = s_k + (i, j)
         i = j
      Improve solution s_k with a local search

   For all arc (i, j) of the best tour s* among s_1, s_2, …, s_m
         τ_ij = max(τ_min , min(τ_max , (1 − ρ)τ_ij + Q/Length(s*) ) )

Return The best tour found
```

MMAS for TSP: Pherome Update

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

For all arc (i, j) of the best tour s^* among $s_1, s_2, \ldots, s_m$

$$\tau_{ij} = \max(\tau_{min}\ ,\ \min(\tau_{max}\ ,\ (1-\rho)\tau_{ij} + Q/Length(s^*)\ )\ )$$

Steiner Tree Problem

Definition: Steiner Tree Problem

Problem Definition:

Given a network $G = (V, E, C)$ with

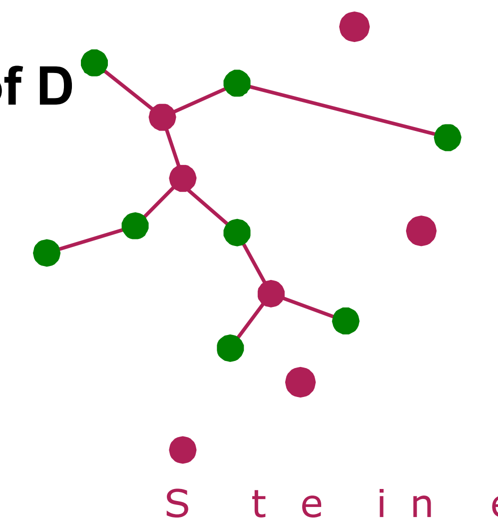
V set of vertices

E set of (undirected) edges

C a function that specifies the weights of the edges

and a subset $D \subseteq V$,

find a tree of minimum weight that contains **all vertices of D** (and may include additional vertices of V not in D).



The optional nodes (which are not in D) are called Steiner nodes.

The nodes in D are called terminals.

Question

Which problem is "easier" to solve?

a) Minimum Spanning Tree

b) Steiner Tree

c) Both are equal

Steiner Tree vs. MST

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

Join at menti.com | use code 8283 7649

Mentimeter

Which problem is "easier" to solve?

SOLUTION: Steiner Tree vs. MST

- If D = V in the Steiner Tree Problem, then we have an MST-Problem. Thus, in general if D ≠ V the Steiner Tree Problem is harder.
- In fact, the Steiner Tree Problem is NP-hard, while MST is solvable in linear time (Prim’s algorithm, with respect to the edges).
- Remark: If |D|=2, then the Steiner Tree Problem is equivalent to the shortest path problem and thus solvable in linear time (Dijkstra’s algorithm, with respect to the edges).

Steiner Tree Problem: Real World Applications

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

- Layout of electrical networks, water distribution networks, etc.: The edges of the input graph typically correspond to segments in the local street map, with vertices representing street intersections and the location of potential customers (terminals). The cost c associated with an edge is the cost of laying the pipe or cable on the corresponding street segment.

- Efficient point to multipoint transfers across multiple datacenters in cloud services (see e.g. https://www.usenix.org/system/files/conference/hotcloud17/hotcloud17-paper-noormohammadpour.pdf)

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

ACO for Steiner Tree Problem

Propose an ACO for the Minimum Steiner Tree problem. Describe how to choose the pheromone trails and how they are exploited.

SOLUTION: ACO for Steiner Tree

IMPORTANT: there is no need in ACO that the "ants" really follow a "consecutive" path

ACO for Steiner Tree Problem

- Element *e* of a partial solution = edge of a tree
- An edge can only be selected as next element if it does not create a cycle
- Probability to chose edge *e* depends on lengths of *e* (in the graph) and on pheromone trail on *e*
- Stop building as soon as all mandatory nodes are in the solution and solution is a tree

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

ACO for Numerical Problems

In graph-based problems such as TSP or Steiner Tree Problem, it is usually obvious what a "step" of an ant is: Choosing an edge to the next city or node.

What about problems that are **not** graph-based, such as the Knapsack Problem? What would a "step" of an ant be here?

SOLUTION: ACO for Knapsack

Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
IAMP Institut für Angewandte Mathematik und Physik

ACO for Numerical Problems

For Knapsack Problem with items a_1, .. a_n:

- Start with empty solution.
- Partial solution is a selection of k items.
- Potential next steps are taking any of the items that are not yet in the knapsack and which still fit in.
- Options: Either use value or ratio of value/weight to select the next item.
- Probability to chose item *a* depends on value (or value/weight) of *a* and on pheromone trail on *a.*

Santa's Challenge