Nature-inspired Algorithms

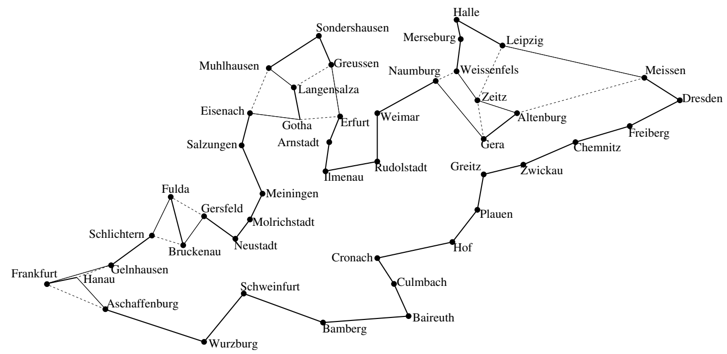
Lecture 5

Algorithm Engineering Group
Hasso Plattner Institute, University of Potsdam

12 June 2017

A historical problem with obscure roots

1832 – Der Handlungsreiſende – wie er ſein ſoll und waſ er zu tun hat, um Aufträge zu erhalten und eineſ glücklichen Erfolgſ in ſeinen Geſchäften gewiß zu ſein – Von einem alten Commiſ-Voyageur



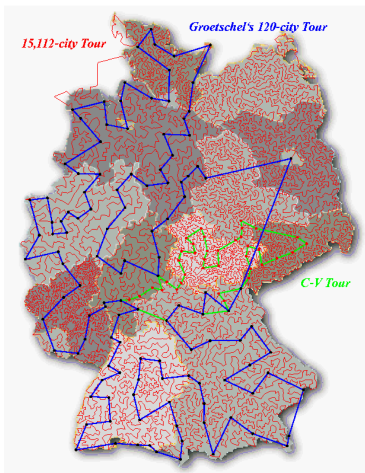
Alexander Schrijver, "On the history of combinatorial optimization (till 1960)", *Handbook of Discrete Optimization*, (K. Aardal, G.L. Nemhauser, R. Weismantel, eds.), Elsevier, Amsterdam, 2005, pp. 1–68.

A historical problem with obscure roots

Karl Menger in 1930: at *mathematisches Kolloquium* in Vienna – *shortest path through a set of points in space*

"[T]his problem is solvable by finitely many trials. **Rules which would push the number of trials below the number of permutations are not known.** The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general does not yield the shortest route."

A historical problem with obscure roots

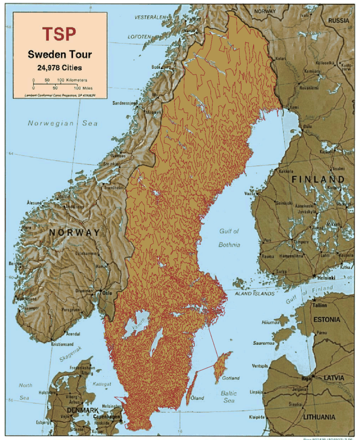


1832 tour

Martin Groetschel, *Mathematical Systems in Economics* (1977)

15 112 city instance (Applegate, Bixby, Chvátal, Cook 2001)

A historical problem with obscure roots



24 978 city instance (Applegate, Bixby, Chvátal, Cook, Helsgaun 2004)

1954: state of the art was 49 city tour (Dantzig et al.)
$24\,978! \approx 9.67 \times 10^{98996}$

TSP

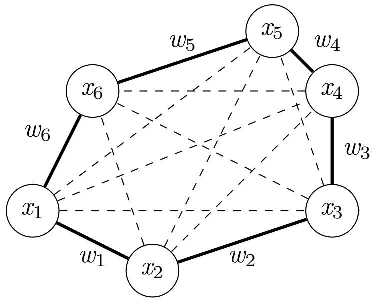
Let $G = (V, E)$ be an undirected complete graph with positive edge weights $w : E \to \mathbb{R}^+$.

- **tour** – a cycle that visits every vertex $v \in V$ exactly once
- **cost** – if x is a tour defined by a sequence of vertices, the cost of x is:

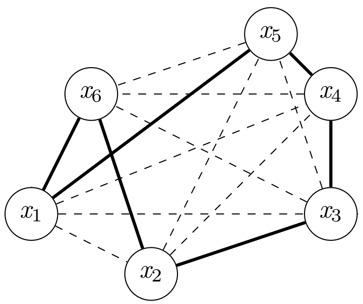
$$f(x) = \sum_{i=1}^{n-1} w(x_i, x_{i+1}) + w(x_1, x_n)$$

$x = (x_1, x_2, x_3, x_4, x_5, x_6)$

$f(x) = w_1 + w_2 + w_3 + w_4 + w_5 + w_6$



Nearest-neighbor heuristic



Can make a bad local decision early

Could lead to an arbitrarily bad tour

Approximating the TSP

TSP is NP-hard: no polynomial time algorithm is known to solve it exactly

Do we have hope to find an approximate solution?

Definition

A ρ-approximation algorithm is an algorithm that guarantees a solution x with $f(x) \leq \rho\mathrm{OPT}$ where OPT is the value of the optimal solution.

Approximating the TSP: edge weights matter!

Arbitrary weights

- $w: E \to \mathbb{R}^+$ arbitrary
- **NP-hard to find a ρ-approximation for any fixed ρ**

Metric

- w satisfies the triangle inequality: $w(a, b) + w(b, c) \geq w(a, c)$.
- **Constant-factor approximation available in polytime**

Euclidean

- Each point $v \in V$ associated with coordinates in some $\mathbb{R}^d$
- $w(u, v)$ is Euclidean distance between u and v
- **Special case of metric**

MST heuristic for metric TSP

input : A graph $G = (V, E)$
output: A tour x
Construct the minimum spanning tree T of G;
Let $x = (x_1, x_2, \ldots, x_n)$ be the nodes in a pre-order traversal of T;

Algorithm 1: ApproxTSPTour

MST heuristic for metric TSP

Theorem.
The ApproxTSPTour algorithm is a 2-approximation algorithm for metric TSP.

Proof.

- Let $x^\star$ be the optimal tour. Let $T^\star$ be the tree obtained by removing one edge from $x^\star$,
- Denote as MST the weight of the minimal spanning tree for G. Then $MST \leq w(T^*) < f(x^\star)$
- If x is created by ApproxTSPTour, then $f(x) \leq 2MST$ (shortcut argument)
- We conclude $f(x) \leq 2MST \leq 2f(x^\star)$
- Time bound

Christofides heuristic for metric TSP

Christofides (1976)

- build the MST T
- find a **min weight perfect matching** M on the vertices of odd degree in T
- construct **Eulerian tour** on multigraph $T \uplus M$
- Remove any repeated vertices

Why does a perfect matching always exist?

Why does an Eulerian tour always exist?

Christofides heuristic for metric TSP

Lemma

Let $S \subseteq V$ be any subset of vertices such that $|S|$ is even. The cost of a perfect matching on the induced subgraph $G[S]$ is at most $f(x^\star)/2$.

Proof.

- Let $x^\star$ be the optimal tour.
- Let x' be the tour formed from $x^\star$ by dropping all the vertices not in S
- Claim: $f(x') \leq f(x^\star)$ **(why?)**
  - Since the graph is metric, shortcutting only reduces total weight
- Claim: there is a matching in $G[S]$ with weight at most $f(x')/2 \leq f(x^\star)/2$ **(why?)**
  - There are two matchings already in x' that add to $f(x')$
- Thus a minimum matching must have weight at most $f(x^\star)/2$

Christofides heuristic for metric TSP

Christofides (1976)

- build the MST T
- find a **min weight perfect matching** M on the vertices of odd degree in T
- construct **Eulerian tour** on $T \cup M$
- Remove any repeated vertices

What is the weight of the Eulerian tour on $T \cup M$?

- Know from MST heuristic: $w(T) \leq f(x^\star)$
- Know from the previous lemma that $w(M) \leq f(x^\star)/2$

Thus, $w(T \cup M) = w(T) + w(M) \leq (1 + 1/2)f(x^\star) = (3/2)f(x^\star)$.

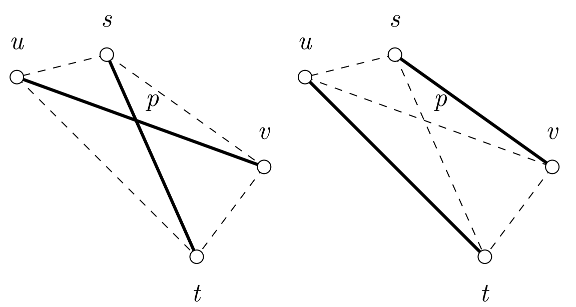
Changing Eulerian tour to Hamilton path only reduces weight (metric)

$\implies$ **Christofides heuristic is a 3/2-approximation for Metric TSP!**

2-opt

Euclidean TSP: Remove edges that cross, and replace with ones that do not

Must improve the tour

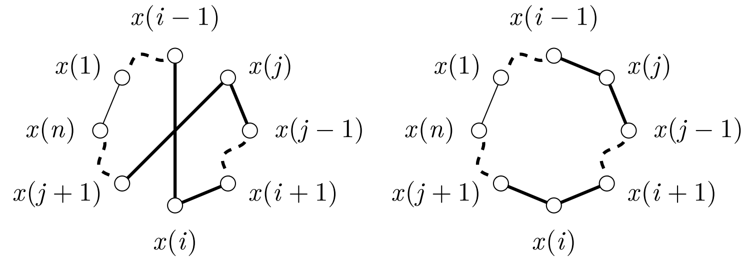


2-opt

Invert a subsequence of the tour from i to j.

$$(x_1, \ldots, x_{i-1}, \underbrace{x_i, x_{i+1}, \ldots x_{j-1}, x_j}_{\text{inversion}}, x_{j+1}, \ldots x_n)$$

$$(x_1, \ldots, x_{i-1}, \overbrace{x_j, x_{j-1}, \ldots x_{i+1}, x_i}, x_{j+1}, \ldots x_n)$$



Local search

Idea

- 2-opt operator generates a "neighborhood" of tours
- iteratively choose the best 2-opt neighbor until no longer possible

Generalizations

- k-opt
  - remove k mutually disjoint edges
  - reassemble remaining fragments into a legal tour
- Lin-Kernighan
  - switch between 2-opt and 3-opt

No performance guarantees currently exist

Can take exponential time to find a local optimum (even on Euclidean instances)[1]

[1] Matthias Englert, Heiko Röglin, and Berthold Vöcking Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP In Proc. of the 18th SODA (New Orleans, USA), pp. 1295-1304, 2007.

Other methods

Idea: every subpath of a path of minimum distance is itself of minimum distance

Held-Karp algorithm (1962)

- Fix a home vertex h. For any $S \subseteq V \setminus \{h\}$,
- $D(S, v) :=$ minimum tour starting at h, ending at v and using only cities in S
- $D(S, v) := \min_{u \in S \setminus \{v\}} D(S \setminus \{v\}, u) + w(u, v)$
- Solution to find $\min_{u \in V \setminus \{h\}} D(V \setminus \{h\}, u) + w(u, h)$
- Dynamic programming for D in time $O(n^2 2^n)$

Still better than n!

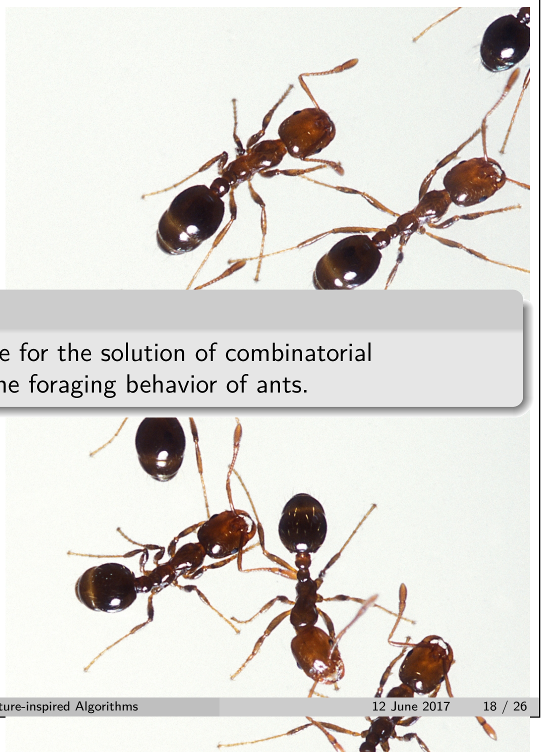
Mixed-integer linear programming

- State TSP as an integer program
- Solve linear relaxation, add further constraints

Tools

CONCORDE: `http://www.math.uwaterloo.ca/tsp/concorde.html`

Ants!

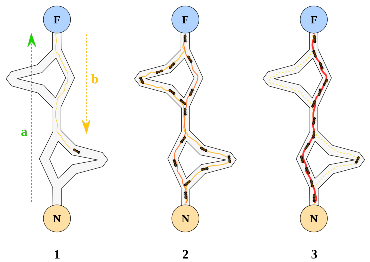


Recall from lecture 2

Ant Colony Optimization

A population-based search technique for the solution of combinatorial optimization problems inspired by the foraging behavior of ants.

Ants!



`http://en.wikipedia.org/wiki/Ant_colony_optimization`

How real ants find shortest paths

- Lay down chemicals **(pheromones)** on the ground: signal to other ants
- Ant decides with some probability to follow a trail, it lays more pheromone, **reinforcing the trail**
- More ants following a trail $\implies$ stronger pheromone signal $\implies$ more ants follow trail in future.
- Pheromone strength decays w/ time
- $\implies$ build-up on shorter paths is faster **(not as much time to decay)**

Using Ants to Find Short Hamiltonian Tours

Still want ants to find a shortest path: but now we need to impose additional constraints!

Ant System for TSP

1. Each ant builds a tour from a starting vertex
2. Each ant at a vertex i chooses a vertex j to visit next with a probability that is a function of
   - distance $w(i, j)$ **(probabilistic Nearest-Neighbor heuristic)**
   - pheromone concentration on edge (i, j)
   - infeasible vertices have probability zero **(enforce Hamiltonian constraint)**
3. When tour is complete, lay pheromone on each visited edge
4. Evaporate pheromone on *all* edges

Construction step

Probability rule

Suppose an ant has built a partial tour on a set of vertices $S \subset V$ ending at vertex i. The probability of choosing city $j \in V \setminus S$ is

$$p(i, j) = \frac{\tau(i, j)^{\alpha} \cdot w(i, j)^{-\beta}}{\sum_{k \in V \setminus S} \tau(i, k)^{\alpha} \cdot w(i, k)^{-\beta}}$$

- $\tau(i, j)$ is the strength of pheromone on edge (i, j)
- $w(i, j)^{-\beta}$ is a *local heuristic* that make ants prefer nearer cities:
- α and β are balancing factors: (constant parameters, e.g., $\alpha = 1$, $\beta = 2$).

τ and $1/w$ can be seen as trading off global (pheromone) information and local (distance) information during tour construction.

After all ants construct tours

Global pheromone evaporation

Suppose the k-th ant constructs a tour of length $L(k)$. Then it adds a pheromone amount of $q/L(k)$ to each edge in its tour*.

$$\Delta_{ij}(k) = \begin{cases} q/L(k) & \text{if } (i,j) \text{ is in the } k\text{-th ant's tour,} \\ 0 & \text{otherwise.} \end{cases}$$

Pheromone trail evaporates a small amount after every iteration

$$\tau(i, j) = \rho\tau(i, j) + \sum_{k=1}^{m} \Delta_{ij}(k)$$

where $0 < \rho < 1$ is an evaporation factor.

* q is a fixed constant

Ant System summary

Initialization: set pheromone strength to small values

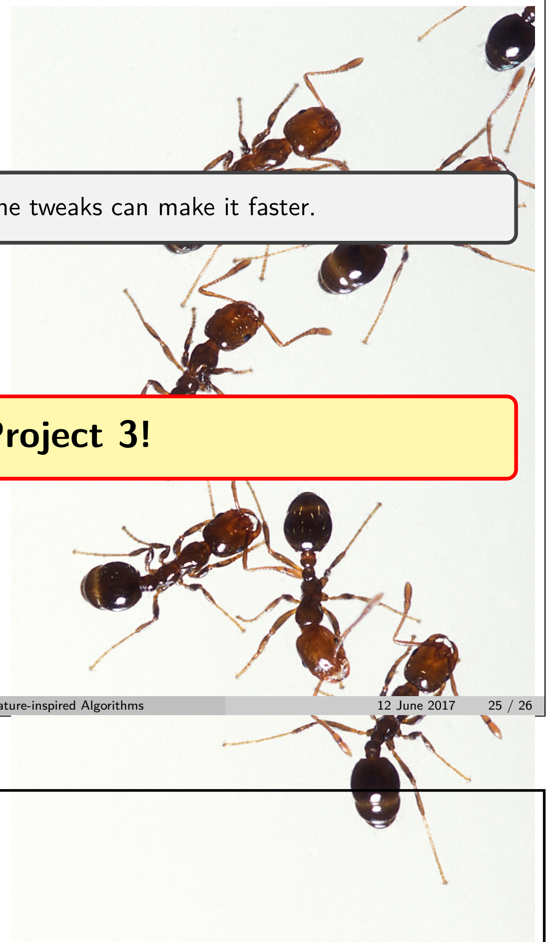
Construction: in each iteration ants build their individual tours and pheromone is laid down when their tours are complete

Termination: iterate until tour counter reaches maximum or all ants are building the same tour

Want pheromone to build up on shortest tour faster than others

Additions

Local search: apply LS (e.g., 2-opt) while building or after building tour. Ends up with shorter tour, might get better solutions faster



This version works OK, but some tweaks can make it faster.

Project 3!

Further reading:

- Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. MIT Press. 2004
- Marco Dorigo, Christian Blum, Ant colony optimization theory: A survey, Theoretical Computer Science, Volume 344, Issue 2, 2005, Pages 243-278. `http://dx.doi.org/10.1016/j.tcs.2005.05.020`
- Timo Kötzing, Frank Neumann, Heiko Röglin, Carsten Witt. *Theoretical analysis of two ACO approaches for the traveling salesman problem.* Swarm Intelligence 2012: 1-21. `http://dx.doi.org/10.1007/s11721-011-0059-7`

Ant-system slides adapted from ACO lecture by Dr Michael Herrmann at University of Edinburgh