

Routing problems

Guillaume Dalle (ENPC, CERMICS)

REOP - Class 9 (15/12/2021)

Contents

1	Traveling Salesperson Problem (TSP)	1
1.1	Heuristics	1
1.1.1	No guarantees	1
1.1.2	Approximation algorithms	1
1.2	Integer Programming	2
1.2.1	Hamiltonian cycle polytope	2
1.2.2	Branch and Cut	2
1.3	Theoretical results	2
2	Postperson Tour Problem	3

We focus on problems with a single vehicle. The input is always the same: a graph $G = (V, E)$ and a cost function $c : E \rightarrow \mathbb{R}_+$. We consider undirected graphs (see textbook for the directed case).

1 Traveling Salesperson Problem (TSP)

Goal: find a cycle C of minimum cost $\sum_{e \in E} c(e)$ that visits each vertex at least once.

Reformulation: find a Hamiltonian cycle on the complete graph K_n with shortest path costs \implies triangular inequality $c(u, w) \leq c(u, v) + c(v, w)$. This is called the metric TSP.

1.1 Heuristics

1.1.1 No guarantees

Greedy: Nearest neighbor

Local search: 2-OPT, k -OPT, Lin-Kernighan

1.1.2 Approximation algorithms

Double tree algorithm: find a MST, double the edges, construct a Eulerian tour, take shortcuts to ignore vertex repetitions \implies approx. ratio of 2

Proof: MST is a lower bound on TSP.

Christofides algorithm: find a MST, find a minimum weight perfect matching between odd-degree vertices, double the edges of the matching, construct a Eulerian tour, take shortcuts to ignore vertex repetitions.

Proof: MST is a lower bound on TSP + each optimal tour is the union of two perfect matchings

Exercise 15.1: Number of vertices of odd degree (handshaking lemma)

1.2 Integer Programming

1.2.1 Hamiltonian cycle polytope

The TSP can be formulated as an Integer Program:

$$\min \sum_{e \in E} c(e)x_e \quad \text{s.t.} \quad \begin{cases} x_e \in \{0, 1\} & \forall e \\ \sum_{e \in \delta(v)} x_e = 2 & \forall v \\ \sum_{e \in \delta(X)} x_e \geq 2 & \forall X \subsetneq V, X \neq \emptyset \end{cases} \quad (\text{TSP})$$

The subtour elimination constraint boils down to finding a minimum cut in the graph $K_n[x]$, which means we can separate it in polynomial time.

Exercise: Prove that the subtour elimination constraint for the TSP can be reformulated as $\sum_{e \in E(X)} x_e \leq |X| - 1$, which we saw in the MST formulation.

Exercise 15.6: TSP with time constraints.

1.2.2 Branch and Cut

The number of constraints is exponential, which is why we cannot include them all in the Branch & Bound scheme. We use a constraint generation method: at each node of the B&B tree, we start with a small number of subsets X and iteratively add constraints that are violated by the solution to the linear relaxation, until we can prove (continuous) feasibility.

In this case, the subproblems are a harder than the root problem: we must solve TSP relaxations where some edges are forbidden and others are imposed.

Exercise: We consider a constrained TSP where the solution must use edges from $A \subset E$ and cannot use edges from set $B \subset E$. Modify the cost function to formulate this as an unconstrained TSP.

The point of this trick is to use standard lower bounds for the unconstrained TSP at every node of the tree (such as the Held-Karp bound, see textbook).

1.3 Theoretical results

Theorem (Beardwood, Halton & Hammersley, 1956): The length L_n of the shortest cycle through n independent uniform random points in $[0, 1]^2$ satisfies $L_n/\sqrt{n} \rightarrow \beta$ almost surely, where β is a constant whose value is not known explicitly.

Theorem (Karp, 1972): The TSP is NP-hard.

Theorem (Sahni & Gonzalez, 1976): It is NP-hard to approximate the general (non-metric) TSP within any constant factor.

Theorem (Christofides, 1976): The metric TSP has a polynomial $\frac{3}{2}$ -approximation algorithm.

Theorem (Ahora, 1998): The Euclidean TSP has a polynomial $(1 + \epsilon)$ -approximation algorithm for any $\epsilon > 0$.

Theorem (Karpinski, Lampis & Schmied, 2015): It is NP-hard to approximate the metric TSP within a factor of $\frac{123}{122}$, and the metric *asymmetric* TSP within a factor of $75/74$.

Theorem (Svensson, Tarnawski & Vég, 2018): The metric *asymmetric* TSP has a polynomial 506-approximation algorithm.

Theorem (Karlin, Klein & Gharan, 2020): The metric TSP has a polynomial $(\frac{3}{2} - 10^{-36})$ -approximation algorithm.

2 Postperson Tour Problem

Goal: find a cycle C of minimum cost $\sum_{e \in E} c(e)$ that visits each edge at least once. G is a city, vertices are intersections, edges are streets where the postperson must leave mail at each door.

Exercise: Find a flow formulation.

If the graph is Eulerian, any Eulerian cycle is optimal.

Lemma 15.6: The optimal postman tour visits each edge at most twice.

Proof idea: A postman tour on G defines a Eulerian multigraph G' , and if e is present $k \leq 3$ times, then removing e two times keeps G' Eulerian, which enables us to rebuild a cheaper postman tour on G .

We look for the cheapest set of edges F to duplicate in order to make G Eulerian.

Theorem 15.7: We can find F in polynomial time.

Proof idea: Same as Christofides algorithm: F is a union of paths linking the vertices of odd degree in G . We match these vertices together using the shortest path distance between them as cost.