The TSP: the general case and the metric case

Outline

Why guarantees, and why limits?

The general TSP is not approximable

A special case: the metric TSP

The double tree and the Christofides algorithms
- The double tree algorithm
- The Christofides algorithm

Structural properties

Performance guarantees
- The worst case performance ratio of DT
- The worst case performance ratio of CH

Sources

Typical mistakes

Learning objectives

At the end of this chapter you will be able to:

- prove that no constant guarantee exists for the general TSP, unless $P = NP$;
- explain why the triangle inequality is what makes the TSP approximable;
- execute the double tree and the Christofides algorithms by hand on a metric instance;
- prove the guarantees 2 of the double tree and $\frac{3}{2}$ of the Christofides algorithm;
- build the families of instances that make these guarantees tight.

Prerequisites

- From this module, the chapter «Principles of approximation algorithms»: performance ratio, relative performance guarantee, tight guarantee and worst-case performance ratio.
- From the module *Combinatorial Optimization Problems*, the chapter «Routing problems»: the definition and the reference instance of the TSP.
- From the module *Network Optimization*, the chapter «Minimum spanning trees»: Kruskal's algorithm
- From the module *Combinatorial Optimization Problems*, the chapter «Combinatorial optimization: definitions»: polynomial time, the classes P and NP, NP-completeness.

Why guarantees, and why limits?

Introduction: Why guarantees, and why limits?

- A heuristic algorithm for the TSP returns a tour, but how far can it be from the shortest one?
- In this chapter we answer for *every* instance, not only for the instances we try
- The answer has two sides: guarantees when they exist, and limits when they do not

Guarantees and limits for the TSP

- The **positive results** of this chapter bound the gap between the heuristic tour and the optimal tour on *every* instance of the metric TSP.
- For the **general TSP** we prove the opposite: unless $P = NP$, no polynomial algorithm can have a good guarantee.
- The **negative results** tell us when we must give up asking.

Summary

Guarantees:

- They hold on every instance, and for the TSP they exist on the metric instances

Limits:

- For the general TSP no polynomial algorithm has a good guarantee, unless $P = NP$

The general TSP is not approximable

Introduction: The general TSP is not approximable

- In this section we recall the TSP and the performance ratio of an algorithm
- Then we prove that, unless $P = NP$, no polynomial algorithm for the general TSP has a constant guarantee
- The proof shows what goes wrong: huge distances, which violate the triangle inequality

The problem of this chapter

- The problem of this chapter is the travelling salesman problem (TSP)
- Its definition and its assumptions are on its page

Instances and tours

- As on the problem page, an instance has a set $\mathscr{V} = \{1, 2, \ldots, n\}$ of cities, $n \geq 3$, and symmetric distances

$$d_{ij} = d_{ji} \in \mathbb{Z}_{\geq 1}$$

- A tour σ is a permutation of $\mathscr{V}$, of length

$$L(\sigma) = \sum_{k=1}^{n-1} d_{\sigma_k \sigma_{k+1}} + d_{\sigma_n \sigma_1}$$

- The length is minimized.

Recall: the performance ratio

- As in the chapter «Principles of approximation algorithms», $z(I)$ is the optimal value of the instance I and $\mathscr{A}(I)$ the value of the solution returned by the algorithm $\mathscr{A}$
- The TSP is a minimization problem, hence for every instance I

$$\mathscr{A}(I) \geq z(I) \qquad \text{and} \qquad \frac{\mathscr{A}(I)}{z(I)} \geq 1$$

- The worst case performance ratio is

$$r(\mathscr{A}) = \sup_I \frac{\mathscr{A}(I)}{z(I)}$$

No constant guarantee

- For the general TSP no polynomial algorithm has a constant guarantee, unless $P = NP$.
- The proof follows the scheme of all such results: an approximation algorithm with a too good guarantee would decide in polynomial time an NP-complete problem.

The general TSP is not approximable

Proposition 1

Let $\alpha \geq 1$ be a constant. If $P \neq NP$, there is no polynomial algorithm $\mathscr{A}$ for the TSP such that, for every TSP instance I,

$$\frac{\mathscr{A}(I)}{z(I)} \leq \alpha.$$

Proof (1/3)

- The **Hamiltonian cycle problem** asks whether a graph $G = (V, E)$ has a cycle through all its vertices; it is NP-complete.
- Suppose that $\mathscr{A}$ exists, and, given a graph G with $n \geq 3$ vertices, build the TSP instance I_G on the cities $\mathscr{V} = V$ with

$$d_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E, \\ \lceil \alpha n \rceil + 1 & \text{otherwise.} \end{cases}$$

Proof (2/3)

- If G has a Hamiltonian cycle, that cycle is a tour of length n, so

$$z(I_G) = n \qquad \text{and} \qquad \mathscr{A}(I_G) \leq \alpha n$$

- If G has no Hamiltonian cycle, every tour uses at least one pair $\{i, j\} \notin E$, so

$$\mathscr{A}(I_G) \;\geq\; z(I_G) \;\geq\; (n-1) + \lceil \alpha n \rceil + 1 \;>\; \alpha n$$

Proof (3/3)

- Hence

$$G \text{ is Hamiltonian} \iff \mathscr{A}(I_G) \leq \alpha n$$

- Building I_G and running $\mathscr{A}$ takes polynomial time, so the Hamiltonian cycle problem would be in P, and $P = NP$.

□

The instances I_G are not metric

- The instance I_G violates the **triangle inequality** $d_{ij} \leq d_{ik} + d_{kj}$ as soon as G has a path i, k, j with $\{i, j\} \notin E$:

$$d_{ij} = \lceil \alpha n \rceil + 1 > 2 = d_{ik} + d_{kj}$$

- The proof needs such huge distances, and it says nothing on the instances that satisfy the inequality.

- The result and its proof are due to Sahni and Gonzalez (1976), and the NP-completeness of the Hamiltonian cycle problem to Karp (1972).

General and metric TSP

- The general TSP is not approximable: unless $P = NP$, no polynomial algorithm has a constant performance guarantee.
- The *metric* TSP, the special case with the triangle inequality, is approximable, and the rest of the chapter is devoted to it.

Summary

Inapproximability:

- An algorithm with a constant guarantee α would decide the Hamiltonian cycle problem in polynomial time
- Hence no such algorithm exists for the general TSP, unless $P = NP$

What the proof needs:

- Distances $\lceil \alpha n \rceil + 1$, which violate the triangle inequality
- Nothing is said on the metric instances

A special case: the metric TSP

Introduction: A special case: the metric TSP

- In this section we define the metric instances of the TSP: those with the triangle inequality
- The triangle inequality is what makes the TSP approximable
- We also recall the reference instance, on which we run the algorithms of the chapter

The metric TSP

Definition 1: metric TSP

An instance of the TSP, with cities $\mathcal{V} = \{1, 2, \ldots, n\}$ *and symmetric distances* $d_{ij} = d_{ji} \in \mathbb{Z}_{\geq 1}$, *is* ***metric*** *if the distances satisfy the triangle inequality*

$$d_{ij} \leq d_{ik} + d_{kj} \qquad \text{for all distinct } i, j, k \in \mathcal{V}.$$

The triangle inequality

- The **metric TSP** is the TSP restricted to the metric instances.
- The instances I_G of the inapproximability proof are not metric.
- The triangle inequality is exactly what makes the TSP approximable, and every guarantee of the next sections uses it.

The reference instance

The reference instance has 5 cities, it is metric, and its unique optimal tour is $(1, 3, 5, 2, 4)$, with $z(I) = 18$. Its distances are:

d_{ij}	1	2	3	4	5
1	—	5	2	2	9
2	5	—	4	3	4
3	2	4	—	4	7
4	2	3	4	—	7
5	9	4	7	7	—

Summary

Metric instances:

- The distances satisfy $d_{ij} \leq d_{ik} + d_{kj}$ for all distinct i, j, k
- The instances I_G of the inapproximability proof are not metric

Reference instance:

- 5 cities, metric, unique optimal tour $(1, 3, 5, 2, 4)$ with $z(I) = 18$

The double tree and the Christofides algorithms

Introduction: The double tree and the Christofides algorithms

- In this section we describe two constructive heuristics for the metric TSP
- Both start from a minimum spanning tree, build a closed walk through all the cities and turn it into a tour
- We run both of them by hand on the reference instance

Minimum spanning tree

- The two algorithms of this section build a tour from a **minimum spanning tree** (MST) of the complete graph on $\mathscr{V}$, with edge weights d_{ij}.
- We compute it with Kruskal's algorithm: the edges are examined by non-decreasing distance (ties by the pair of indices) and an edge is taken if it does not close a cycle with the edges already taken.
- The weight of a tree T is

$$d(T) = \sum_{\{i,j\} \in T} d_{ij}$$

Shortcutting

- Both algorithms first build a closed walk that passes through every city, possibly more than once.
- Then they turn it into a tour by **shortcutting**: the walk is followed and every city already visited is skipped.
- In other words, a city is kept only at its first occurrence.

Eulerian circuits and Fleury's rule

- A closed walk that uses every edge of a (multi)graph exactly once is an **Eulerian circuit**.
- In the examples we build it with **Fleury's rule**:
  - start from city 1;
  - at each step, leave the current city along an unused edge towards the neighbour of smallest index,
  - unless that edge is a *bridge* of the unused edges (its removal disconnects them) and another edge is available.

The double tree algorithm

- The *Double Tree Algorithm* (denoted by DT) is a constructive heuristic for the metric TSP.
- It computes a minimum spanning tree T, doubles each of its edges, follows an Eulerian circuit of the doubled tree, and shortcuts it into a tour.

Algorithm: Double Tree

Input : metric TSP instance with n cities and distances $d[1..n][1..n]$

Output: a tour σ and its length $L(\sigma)$

```
1 function DoubleTree(n, d)
2 |   T := a minimum spanning tree of the complete
  |    graph (Kruskal)
3 |   D := the multigraph with two copies of each edge of
  |    T
4 |   W := an Eulerian circuit of D from city 1 (Fleury)
5 |   σ := the cities of W in order of first occurrence
  |    // shortcutting
6 |   return σ, L(σ)
7 end
```

How the double tree algorithm works

- In the doubled tree, Fleury's rule never goes back to the parent while a child has still to be visited, because the last copy of the edge towards the parent is a bridge.
- The Eulerian circuit is therefore a depth-first visit of T from city 1, with the children in increasing order.
- The tour σ is the order in which the visit meets the cities for the first time (the *preorder*).
- With Kruskal's algorithm and a depth-first visit, DT runs in $O(n^2 \log n)$ time.

Example: DT on the reference instance (1/4)

Kruskal's algorithm examines:

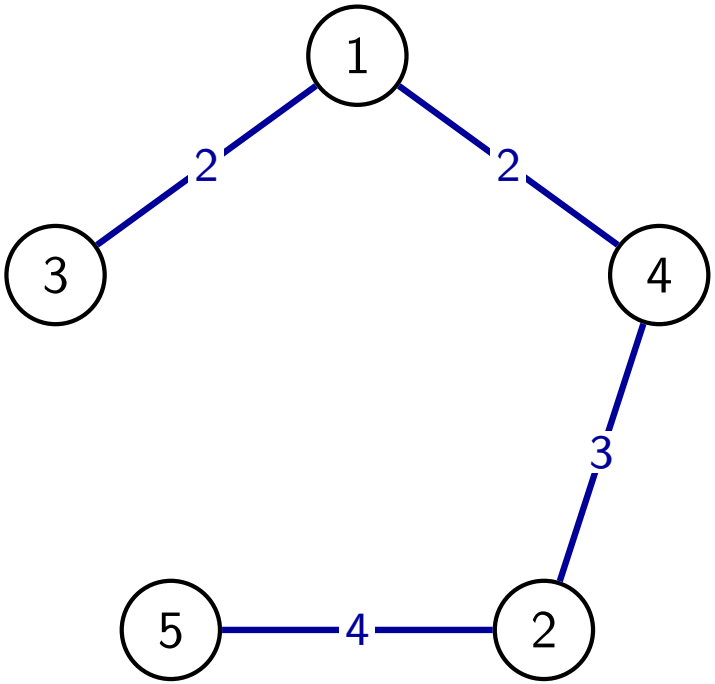
- $\{1,3\}$ (2, taken)
- $\{1,4\}$ (2, taken)
- $\{2,4\}$ (3, taken)
- $\{2,3\}$ (4, it closes the cycle $2,3,1,4$: rejected)
- $\{2,5\}$ (4, taken)

The tree is

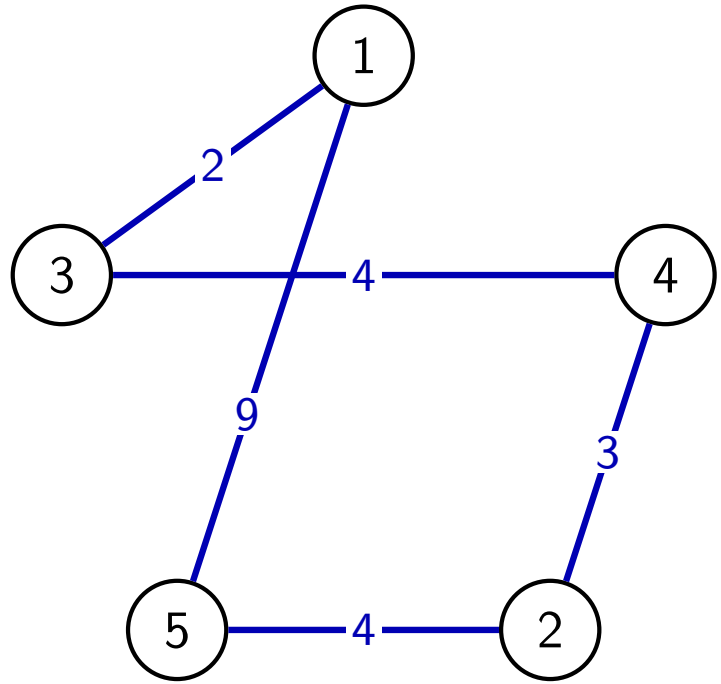
$$T = \{\{1,3\},\{1,4\},\{2,4\},\{2,5\}\}, \qquad d(T) = 2+2+3+4 = 11,$$

and it is the unique MST of the instance.

Example: DT on the reference instance (2/4)



MST T, $d(T) = 11$



tour of DT, $L(\sigma) = 22$

Example: DT on the reference instance (3/4)

- The Eulerian circuit of the doubled tree from city 1 is

$$W = (1, 3, 1, 4, 2, 5, 2, 4, 1),$$

of length

$$2 + 2 + 2 + 3 + 4 + 4 + 3 + 2 = 22 = 2\, d(T)$$

- Keeping each city at its first occurrence gives

$$\sigma = (1, 3, 4, 2, 5),$$

of length

$$L(\sigma) = d_{13} + d_{34} + d_{42} + d_{25} + d_{51} = 2 + 4 + 3 + 4 + 9 = 22$$

Example: DT on the reference instance (4/4)

- The shortcuts replace
  - $3 \to 1 \to 4$ $(2+2)$ by $3 \to 4$ (4),
  - $5 \to 2 \to 4 \to 1$ $(4+3+2)$ by $5 \to 1$ (9).
- On this instance both triangle inequalities used hold with equality, and shortcutting saves nothing.
- Hence

$$DT(I) = 22 \qquad \text{and} \qquad \frac{DT(I)}{z(I)} = \frac{22}{18} = \frac{11}{9}$$

The Christofides algorithm

- The *Christofides Algorithm* (denoted by CH) is a constructive heuristic for the metric TSP.
- It computes a minimum spanning tree T and, instead of doubling it, adds to T a minimum weight perfect matching M on the cities of odd degree in T.
- Then it follows an Eulerian circuit of $T \cup M$ and shortcuts it into a tour.

Algorithm: Christofides

Input : metric TSP instance with n cities and distances $d[1..n][1..n]$

Output : a tour σ and its length $L(\sigma)$

```
1 function Christofides(n, d)
2     T := a minimum spanning tree of the complete graph (Kruskal)
3     O := { v ∈ V : v has odd degree in T } // |O| is even
4     M := a minimum weight perfect matching of the cities in O
5     D := T ∪ M          // multigraph, all degrees even
6     W := an Eulerian circuit of D from city 1 (Fleury)
7     σ := the cities of W in order of first occurrence
          // shortcutting
8     return σ, L(σ)
9 end
```

Why the algorithm is well defined

- **Why $|O|$ is even.** The degrees of a graph sum up to twice the number of edges, an even number, so the number of vertices of odd degree is even, and a perfect matching of O exists (the graph is complete).

- **Why $T \cup M$ has even degrees.** Each city of O gets exactly one more edge from M, and the other cities none. If an edge belongs to both T and M, D contains two copies of it.

Computing the matching

- A minimum weight perfect matching can be computed in polynomial time, $O(n^3)$, with Edmonds' blossom algorithm.
- In the examples and exercises $|O| \leq 4$ and the matchings are enumerated ($|O| = 4$ gives 3 perfect matchings).

Example: *CH* on the reference instance (1/3)

- The MST is the one of the *DT* example, with degrees

$$\deg(1) = 2, \quad \deg(2) = 2, \quad \deg(3) = 1,$$
$$\deg(4) = 2, \quad \deg(5) = 1$$

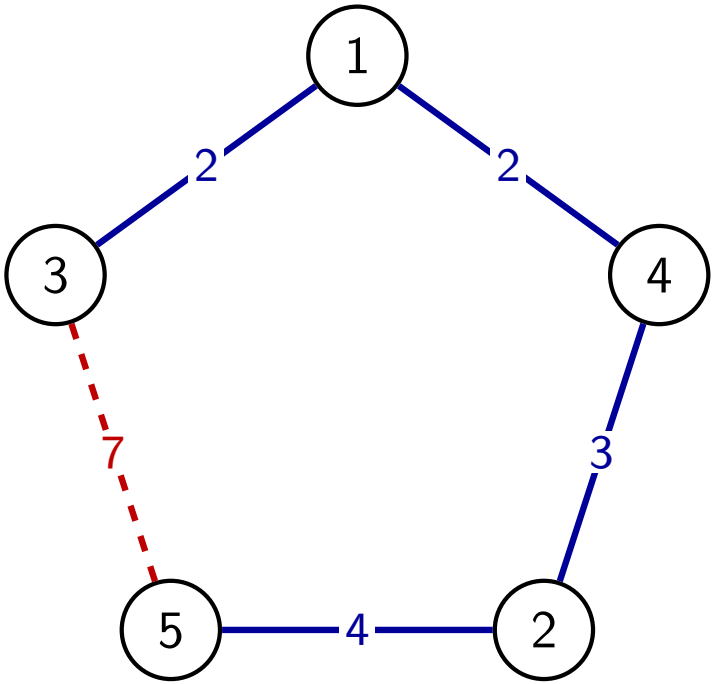
- Hence

$$O = \{3, 5\},$$

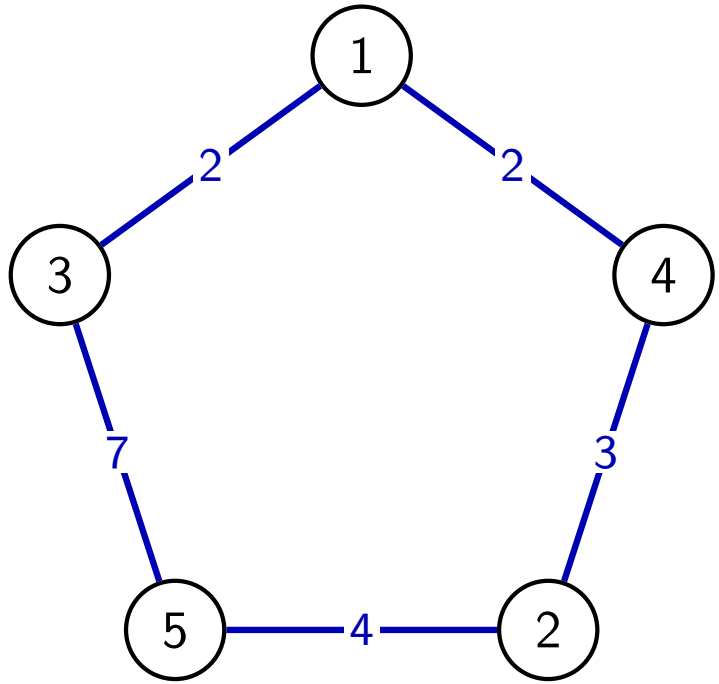
and the only perfect matching is

$$M = \{\{3, 5\}\}, \qquad d(M) = 7$$

Example: *CH* on the reference instance (2/3)



T (solid) and M (dashed)



tour of CH, $L(\sigma) = 18$

Example: *CH* on the reference instance (3/3)

- $T \cup M$ is a cycle through all the cities.
- Fleury's rule from city 1 (neighbours 3 and 4) goes to 3 and gives

$$W = (1, 3, 5, 2, 4, 1), \qquad d(T) + d(M) = 11 + 7 = 18,$$

in which no city is repeated: no shortcut is needed and

$$\sigma = (1, 3, 5, 2, 4), \qquad CH(I) = 18 = z(I)$$

- On the reference instance *CH* finds the optimal tour, while *DT* is 4 units above it.

Summary

Common steps:

- A minimum spanning tree T (Kruskal), an Eulerian circuit (Fleury), shortcutting

Double tree DT:

- Doubles the edges of T; on the reference instance $DT(I) = 22$

Christofides CH:

- Adds a minimum perfect matching M of the odd-degree cities; on the reference instance $CH(I) = 18 = z(I)$

Structural properties

Introduction: Structural properties

- The guarantees of the next section compare the length of a heuristic tour with a lower bound on $z(I)$.
- We need three properties:
  - shortcutting does not increase the length;
  - an MST is shorter than any tour;
  - so is (twice) a minimum perfect matching of an even set of cities.

Shortcutting does not increase the length

Observation 1

In a metric TSP instance, let W be a closed walk that visits every city at least once, and let σ be the tour obtained from W by shortcutting. Then $L(\sigma) \leq$ length of W.

Proof

- Each edge $\{\sigma_k, \sigma_{k+1}\}$ of the tour (and the closing edge $\{\sigma_n, \sigma_1\}$) replaces a stretch $\sigma_k = u_0, u_1, \ldots, u_q = \sigma_{k+1}$ of W whose inner cities are skipped.

- Applying the triangle inequality $q - 1$ times,

$$d_{u_0 u_q} \leq d_{u_0 u_1} + d_{u_1 u_q} \leq \ldots \leq \sum_{p=0}^{q-1} d_{u_p u_{p+1}}$$

- The stretches are disjoint parts of W, so summing over all the edges of σ gives

$$L(\sigma) \leq \text{length of } W$$

□

The MST is a lower bound

Observation 2

For every TSP instance (metric or not) and every minimum spanning tree T,

$$d(T) \leq z(I).$$

Proof

- Removing any edge from an optimal tour leaves a path through all the cities, which is a spanning tree.
- Its weight is $z(I)$ minus the removed distance, at most $z(I)$ since distances are positive.
- The MST weighs no more than any spanning tree.

□

The matching is at most half the optimum

Observation 3

In a metric TSP instance, let $O \subseteq \mathcal{V}$ with $|O|$ even and $|O| \geq 2$, and let M be a minimum weight perfect matching of O. Then

$$d(M) \leq \frac{z(I)}{2}.$$

Proof (1/2)

- Let $\tilde{\sigma}$ be an optimal tour and shortcut it to the cities of O: we obtain a cycle

$$o_1, o_2, \ldots, o_{|O|}, o_1$$

through the cities of O which, by Observation 1, has length at most $z(I)$.

- Since $|O|$ is even, its edges split into two perfect matchings of O, taking alternately one edge each:

$$M_1 = \{\{o_1, o_2\}, \{o_3, o_4\}, \ldots\}$$

$$M_2 = \{\{o_2, o_3\}, \{o_4, o_5\}, \ldots, \{o_{|O|}, o_1\}\}$$

Proof (2/2)

- Hence
$$d(M_1) + d(M_2) \leq z(I),$$
and one of the two weighs at most $\frac{z(I)}{2}$, so
$$d(M) \leq \min\{d(M_1), d(M_2)\} \leq \frac{z(I)}{2}$$
- For $|O| = 2$ the cycle is the edge $\{o_1, o_2\}$ travelled twice and $M_1 = M_2$.

□

Eulerian circuits

- The last property we need is classical: a connected multigraph in which every vertex has even degree has an Eulerian circuit (a classical result of Euler), and Fleury's rule finds one.
- The doubled tree of DT and the multigraph $T \cup M$ of CH are connected (they contain T) and have even degrees.

The properties on the reference instance

- The minimum spanning tree:

$$d(T) = 11 \leq 18$$

- With $O = \{3, 5\}$ the optimal tour shortcut to O is $3, 5, 3$, of length $14 \leq 18$, so

$$d(M) = 7 \leq 9$$

Summary

Shortcutting:

- In a metric instance it does not increase the length of a closed walk

Lower bounds on $z(I)$:

- $d(T) \leq z(I)$ for every minimum spanning tree T
- $d(M) \leq \frac{z(I)}{2}$ for a minimum perfect matching M of an even set of cities

Eulerian circuits:

- They exist in connected multigraphs with even degrees

Performance guarantees

Introduction: Performance guarantees

- In this section we prove the guarantees 2 of DT and $\frac{3}{2}$ of CH, with the structural properties
- Then we build families of instances on which the ratios get as close as we like to 2 and to $\frac{3}{2}$
- Hence both guarantees are tight

The guarantee of DT

Proposition 2

For every metric TSP instance I,

$$\frac{DT(I)}{z(I)} \leq 2.$$

Proof

- The Eulerian circuit W of the doubled tree uses every edge of T twice, so its length is $2\,d(T)$.
- It visits every city because T is spanning.
- By Observation 1 and Observation 2,

$$DT(I) = L(\sigma) \;\leq\; 2\,d(T) \;\leq\; 2\,z(I).$$

□

Is the guarantee 2 tight?

The bound 2 holds on every metric instance.

We now build a family of metric instances on which the ratio of DT tends to 2.

The guarantee of DT is tight

Proposition 3

There are metric TSP instances I_n, $n \geq 5$ odd, such that

$$\frac{DT(I_n)}{z(I_n)} = \frac{2n}{n+2} \rightarrow 2 \qquad \text{as} \qquad n \rightarrow +\infty.$$

Proof (1/4)

- Let $n = 2k + 1$ with $k \geq 2$, $a := n$ and $b := n + 1$.
- City 1 is a centre and the cities $2, \ldots, n$ are leaves; the leaves are joined by the path

$$P = (3, 5, \ldots, 2k + 1, 2, 4, \ldots, 2k),$$

whose consecutive cities never have consecutive indices (the differences are 2, and $2k - 1 \geq 3$ for the edge $\{2k + 1, 2\}$).

Proof (2/4)

- The distances are

$$d_{ij} = \begin{cases} a & \text{if } i = 1 \text{ or } j = 1, \\ b & \text{if } \{i, j\} \text{ is an edge of } P, \\ 2a & \text{otherwise.} \end{cases}$$

- All distances lie between a and $2a$, so

$$d_{ij} \leq 2a \leq d_{ik} + d_{kj} :$$

the instance is metric.

Proof (3/4)

DT.

- The only distances equal to a are those from the centre, and a spanning tree has $n-1$ edges, each of weight at least a: the star centred at 1 is the unique MST, with $d(T) = (n-1)a$.

- The depth-first visit from city 1 meets the leaves in increasing order, $\sigma = (1, 2, 3, \ldots, n)$, and no pair of consecutive leaves is an edge of P:

$$DT(I_n) = a + (n-2)\,2a + a = 2(n-1)a = 2n(n-1)$$

Proof (4/4)

Optimum.

- A tour has exactly two edges at city 1 (length a each) and $n-2$ edges between leaves (length at least b each), so $z(I_n) \geq 2a + (n-2)b$.
- The tour $(1, 3, 5, \ldots, 2k+1, 2, 4, \ldots, 2k)$ follows P and attains this value, hence

$$z(I_n) = 2a + (n-2)b = 2n + (n-2)(n+1) = (n+2)(n-1)$$

- Therefore

$$\frac{DT(I_n)}{z(I_n)} = \frac{2n(n-1)}{(n+2)(n-1)} = \frac{2n}{n+2} = \frac{2}{1+\frac{2}{n}} \rightarrow 2$$

□

The worst case performance ratio of DT

From Proposition 2 and Proposition 3 it follows that DT is a 2-approximation algorithm for the metric TSP, and that its worst case performance ratio is

$$r(DT) = 2$$

Remarks on DT

- For $n = 5$: $a = 5$, $b = 6$, $P = (3, 5, 2, 4)$, $DT(I_5) = 40$, $z(I_5) = 28$ and the ratio is $\frac{10}{7}$.
- For $n = 7$ and $n = 9$ the ratios are $\frac{14}{9}$ and $\frac{18}{11}$.
- On the reference instance $DT(I) = 22 = 2\,d(T)$: the upper bound of the proof is attained, but $z(I) = 18 > d(T) = 11$, and the ratio is only $\frac{11}{9}$.
- The analysis of DT is due to Rosenkrantz, Stearns and Lewis (1977).

The guarantee of *CH*

Proposition 4

For every metric TSP instance I,

$$\frac{CH(I)}{z(I)} \leq \frac{3}{2}.$$

Proof

- The Eulerian circuit W of $T \cup M$ uses every edge once, so its length is $d(T) + d(M)$, and it visits every city.
- We use Observation 1, Observation 2 and Observation 3, with the set O of odd-degree cities ($|O| \geq 2$ because a tree with at least two vertices has at least two leaves):

$$CH(I) = L(\sigma) \leq d(T) + d(M) \leq z(I) + \frac{z(I)}{2} = \frac{3}{2} z(I).$$

□

The guarantee of *CH* is tight

Proposition 5

There are metric TSP instances I_n, $n \geq 5$ odd, such that

$$\frac{CH(I_n)}{z(I_n)} \rightarrow \frac{3}{2} \qquad as \qquad n \rightarrow +\infty.$$

Proof (1/4)

- Let $n \geq 5$ be odd, $a := n$ and $b := n + 1$, so that $a < b < 2a$.
- Consider the graph H on the cities $1, \ldots, n$ with the edges $\{i, i + 1\}$ of length a and $\{i, i + 2\}$ of length b (a strip of triangles).
- Let d_{ij} be the length of a shortest path from i to j in H; shortest path distances always satisfy the triangle inequality.

Proof (2/4)

- A path from i to $j > i$ advances by 1 (cost a) or by 2 (cost b) at each edge.
- Since $b < 2a$ it is best to advance by 2 as much as possible:

$$d_{ij} = \left\lfloor \frac{|i-j|}{2} \right\rfloor b + \left(|i-j| \bmod 2\right) a$$

Proof (3/4)

CH.

- The distances equal to a are exactly those between consecutive cities, so, as in the previous proof, the path $T = \{\{1,2\},\{2,3\},\ldots,\{n-1,n\}\}$ is the unique MST, with $d(T) = (n-1)a$.
- Its odd-degree cities are $O = \{1,n\}$, so $M = \{\{1,n\}\}$, with $d(M) = \frac{n-1}{2}b$.
- $T \cup M$ is the cycle $1,2,\ldots,n,1$, Fleury's rule follows it from city 1 towards city 2, and no shortcut is needed:

$$CH(I_n) = (n-1)a + \frac{n-1}{2}\,b = (n-1)\,\frac{3n+1}{2}$$

Proof (4/4)

An upper bound on the optimum.

- The tour $(1, 3, 5, \ldots, n, n-1, n-3, \ldots, 2)$ uses $n-2$ edges of length b and the two edges $\{n, n-1\}$ and $\{2, 1\}$ of length a, so

$$z(I_n) \leq (n-2)b + 2a = (n-2)(n+1) + 2n = (n+2)(n-1)$$

- Together with Proposition 4,

$$\frac{3}{2} \geq \frac{CH(I_n)}{z(I_n)} \geq \frac{(n-1)(3n+1)}{2(n+2)(n-1)} = \frac{3n+1}{2n+4} = \frac{3+\frac{1}{n}}{2+\frac{4}{n}} \to \frac{3}{2}$$

□

The worst case performance ratio of *CH*

From Proposition 4 and Proposition 5 it follows that *CH* is a $\frac{3}{2}$-approximation algorithm for the metric TSP, and that its worst case performance ratio is

$$r(CH) = \frac{3}{2}$$

Remarks on *CH*

- For $n = 5, 7, 9$ the enumeration of all the tours shows that the upper bound on $z(I_n)$ is the optimum (28, 54, 88), and the ratios are

$$\frac{32}{28} = \frac{8}{7}, \qquad \frac{66}{54} = \frac{11}{9}, \qquad \frac{112}{88} = \frac{14}{11}$$

- The algorithm was found independently by Christofides (1976) and Serdyukov (1978).

- For more than forty years no polynomial algorithm with a better constant was known for the metric TSP; the first improvement, by an extremely small constant, is a randomized algorithm by Karlin, Klein and Oveis Gharan (2021).

Summary

Double tree:

- $DT(I) \leq 2\, d(T) \leq 2\, z(I)$, and the star instances give ratios $\frac{2n}{n+2} \to 2$: $r(DT) = 2$

Christofides:

- $CH(I) \leq d(T) + d(M) \leq \frac{3}{2}\, z(I)$, and the strip of triangles gives ratios $\frac{3n+1}{2n+4} \to \frac{3}{2}$: $r(CH) = \frac{3}{2}$

Sources

Introduction: Sources

- The results of this chapter come from five research papers and reports
- They are the places where to read the original analyses of DT, of CH and of the general TSP

Sources (1/2)

- D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis II, An analysis of several heuristics for the traveling salesman problem, *SIAM Journal on Computing* 6(3):563–581, 1977.
- N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388, Carnegie-Mellon University, 1976.
- A.I. Serdyukov, On some extremal walks in graphs, *Upravlyaemye Sistemy* 17:76–79, 1978.

Sources (2/2)

- S. Sahni, T. Gonzalez, P-complete approximation problems, *Journal of the ACM* 23(3):555–565, 1976.
- R.M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, Plenum Press, 85–103, 1972.

Summary: Sources

- Rosenkrantz, Stearns and Lewis (1977): the analysis of DT
- Christofides (1976) and Serdyukov (1978): the algorithm CH
- Sahni and Gonzalez (1976): the general TSP is not approximable
- Karp (1972): the NP-completeness of the Hamiltonian cycle problem

Typical mistakes

Introduction: Typical mistakes

- Three mistakes that come up often when the algorithms of this chapter are applied
- They concern the triangle inequality, the shortcutting step and the matching of CH
- Checking them is a quick way to avoid losing points

Observation 4: typical mistakes

- ***Applying** DT **or** CH **to a non-metric instance.** The guarantees 2 and $\frac{3}{2}$ use the triangle inequality, in the shortcutting step; without it no constant guarantee exists.*
- ***Shortcutting a city that is visited for the first time.** Following the Eulerian circuit, a city is skipped only if it has already been visited; the first occurrence is kept.*
- ***Matching the wrong cities in** CH. The perfect matching is computed on the cities of odd degree in the spanning tree, whose number is even, and has minimum weight.*

Summary: Typical mistakes

- The guarantees hold only on metric instances
- Shortcutting skips only the cities already visited
- The matching of CH is a minimum perfect matching of the odd-degree cities

Summary (1/3)

- The general TSP is not approximable: an algorithm with a constant guarantee would decide the Hamiltonian cycle problem in polynomial time.
- The metric TSP is the special case that approximates. The guarantees of DT and CH are proved with lower bounds on $z(I)$: the minimum spanning tree and half of the minimum perfect matching; shortcutting, which uses the triangle inequality, turns the Eulerian circuit into a tour that is not longer.
- Both ratios are tight, with families of instances built in this chapter.

Summary (2/3)

problem	algorithm	worst case ratio
general TSP	any polynomial	no constant, unless $P = NP$
metric TSP	DT	2 (tight)
metric TSP	CH	$\frac{3}{2}$ (tight)

Summary (3/3)

Dashboard of the reference instance. The bounds on $z(I)$ found so far:

Instance	Lower bound	Heuristic values	Optimal value
metric TSP, 5 cities	11, minimum spanning tree	DT 22, CH 18	18