

Routing problems

Outline

Why the travelling salesman problem?

Overview of the family

The travelling salesman problem

Counting the tours

A first tour by hand

A combinatorial lower bound

Variants: metric and asymmetric TSP

Remarks

Typical mistakes

In summary

Learning objectives

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

- ▶ state the TSP, the asymmetric TSP and the metric TSP, and describe an instance and a tour;
- ▶ compute the length of a tour, and count the tours of an instance;
- ▶ explain why listing all the tours is hopeless already for 20 cities;
- ▶ build a first tour by hand with the nearest neighbour rule;
- ▶ compute a lower bound on the optimal length from the two shortest edges of every city, and measure the gap with a tour.

Prerequisites

- ▶ From this module, the chapter «Combinatorial optimization problems»: what a combinatorial optimization problem is, how the number of solutions explodes, and the notions of primal bound, dual bound and gap.
- ▶ From the same chapter, the reference TSP instance with 5 cities, which this chapter studies in detail.

Why the travelling salesman problem?

Introduction: Why the travelling salesman problem?

- ▶ We start from a concrete question: in which order should a courier visit the cities of its route?
- ▶ This question is the travelling salesman problem, one of the most studied problems of combinatorial optimization
- ▶ We meet a challenge to try by hand, the records of the exact algorithms, and the reason why the problem is hard

A courier and its route

- ▶ A courier leaves the depot, must deliver a parcel to each of the other cities and come back to the depot.
- ▶ In which order should the cities be visited, so that the total distance travelled is the smallest possible?
- ▶ This is the **travelling salesman problem**.

A challenge

- ▶ Before reading on, look at the distances of the reference instance with 5 cities and try to find a short tour of the 5 cities.

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	—

- ▶ Then ask yourself how you could prove that no shorter tour exists.
- ▶ The chapter answers both questions.

The test bench of exact algorithms

- ▶ The TSP is one of the most studied problems of combinatorial optimization, and it has long been the test bench of exact algorithms.
- ▶ The largest instance solved to optimality, called pla85900, has 85 900 points, which come from a chip design problem.
- ▶ It was solved in 2005–2006 with the Concorde code, and the optimal tour, of length 142 382 641, was certified in a paper of 2009 by Applegate, Bixby, Chvátal, Cook, Espinoza, Goycoolea and Helsgaun.

How hard is the TSP?

- ▶ On the road maps of real countries, the record of 2025 is the instance korea81998, with 81 998 bars of South Korea.
- ▶ These results need very large computations; the instances of this chapter are tiny.
- ▶ The TSP is strongly NP-hard (see the chapter «Combinatorial optimization problems»), so no efficient exact algorithm is expected.
- ▶ Approximation algorithms, such as that of Christofides (1976), are studied later in the module.

The travelling salesman problem:

- ▶ visit all the cities and come back to the start, travelling the smallest total distance
- ▶ two questions: find a short tour, and prove that no shorter tour exists

A famous and hard problem:

- ▶ the record instances (pla85900, korea81998) need very large computations
- ▶ strongly NP-hard: no efficient exact algorithm is expected

Overview of the family

Introduction: Overview of the family

- ▶ The routing problems of this chapter share the same data and the same kind of solution
- ▶ We describe what they have in common: cities, distances, tours and the length of a tour
- ▶ Then we list the members of the family, which differ in the properties of the distances

Data, solution and objective

- ▶ The **routing problems** of this chapter share the same data: a set $\mathcal{V} = \{1, 2, \dots, n\}$ of n cities and, for each pair of distinct cities $i, j \in \mathcal{V}$, a *distance* $d_{ij} \in \mathbb{Z}_{\geq 1}$.
- ▶ A solution is a **tour**, a permutation σ of the cities read as a closed route that visits every city exactly once; its **length** $L(\sigma)$ is the sum of the distances travelled.
- ▶ The objective is to minimize the length of the tour.

Routing problems in one sentence

A routing problem finds a shortest tour through all the cities.

The members of the family differ in the properties of the distances: symmetric, metric or asymmetric.

The members of the family

- The table lists the members of the family, with the section that defines each of them.

Problem	Acronym	Distances	Section
travelling salesman problem	TSP	symmetric, $d_{ij} = d_{ji}$	The travelling salesman problem
metric travelling salesman problem	metric TSP	symmetric, with the triangle inequality	Variants: metric and asymmetric TSP
asymmetric travelling salesman problem	ATSP	d_{ij} may differ from d_{ji}	Variants: metric and asymmetric TSP

Summary

What the routing problems share:

- ▶ data: n cities and the distances d_{ij} between them
- ▶ solution: a tour, a closed route that visits every city exactly once
- ▶ objective: minimize the length $L(\sigma)$ of the tour

What makes them differ:

- ▶ the distances: symmetric (TSP), with the triangle inequality (metric TSP), asymmetric (ATSP)

The travelling salesman problem

Introduction: The travelling salesman problem

- ▶ In this section we define the TSP: its data, an instance, a tour and the length of a tour
- ▶ The TSP is the reference problem of the family: the metric TSP and the asymmetric TSP are defined from it
- ▶ We state the assumptions, and we meet the instance with 5 cities that we follow through the whole chapter

Definition 1: Travelling salesman problem

Given:

1. a set $\mathcal{V} = \{1, 2, \dots, n\}$ of n cities,
2. for each pair of distinct cities $i, j \in \mathcal{V}$, a distance $d_{ij} \in \mathbb{Z}_{\geq 1}$, with $d_{ij} = d_{ji}$,

the **travelling salesman problem (TSP)** asks to find a closed route that starts from a city, visits every other city exactly once and returns to the starting city, so that the total distance travelled is minimized.

TSP instance

► A **TSP instance** is specified by:

1. one integer number:

the number of cities $n \in \mathbb{Z}_{\geq 3}$,

2. one symmetric matrix of integer numbers of size $n \times n$:

$D = (d_{ij})$, containing the distances; the diagonal entries are not used.

TSP feasible solution: the tour

- ▶ A **TSP feasible solution**, called a **tour**, is a permutation of the cities

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n),$$

read as the closed route $\sigma_1 \rightarrow \sigma_2 \rightarrow \dots \rightarrow \sigma_n \rightarrow \sigma_1$.

- ▶ Its **length** is

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

where the last term is the distance travelled to return to the starting city.

Many permutations, one tour

- ▶ Different permutations may describe the same tour:
 - ▶ the starting city can be any of the n cities;
 - ▶ since the distances are symmetric, the tour can be travelled in either direction.

We can now state the TSP formally, as an optimization problem over the permutations.

The TSP stated formally

Given a TSP instance, the TSP can be formally stated as:

$$\tilde{\sigma} \in \arg \min_{\sigma} \left\{ L(\sigma) = \sum_{k=1}^{n-1} d_{\sigma_k \sigma_{k+1}} + d_{\sigma_n \sigma_1} : \right. \\ \left. \sigma \text{ is a permutation of } \mathcal{V} \right\}.$$

- We denote by $z(I)$ the length of an optimal tour $\tilde{\sigma}$ of the instance I .

Assumptions

► We make the following assumptions:

1. There are at least three cities (with one or two cities there is only one route). Formally:

$$n \geq 3$$

2. The distances are symmetric (travelling from i to j costs as much as travelling from j to i). Formally:

$$d_{ij} = d_{ji}, \quad \forall i, j \in \mathcal{V}, \quad i \neq j$$

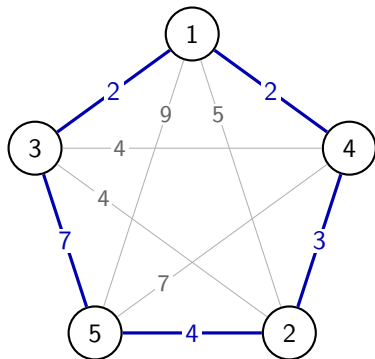
We now look at a TSP instance with 5 cities:
it is the reference instance of the whole chapter.

Example: TSP instance (1/2)

We consider a set $\mathcal{V} = \{1, 2, 3, 4, 5\}$ of 5 cities ($n = 5$), with the following distances:

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	—

Example: TSP instance (2/2)



Its length is

$$L(\sigma) = d_{13} + d_{35} + d_{52} + d_{24} + d_{41} = 2 + 7 + 4 + 3 + 2 = 18.$$

The instance can be drawn as a complete graph:

- ▶ one node per city
- ▶ one edge per pair of cities, labelled with the distance
- ▶ the bold edges form the tour $\sigma = (1, 3, 5, 2, 4)$
- ▶ the cities are drawn in the order of the tour

Summary

The TSP:

- ▶ $n \geq 3$ cities and symmetric integer distances $d_{ij} = d_{ji}$
- ▶ a tour is a permutation σ of the cities, read as a closed route
- ▶ its length $L(\sigma)$ includes the return distance $d_{\sigma_n \sigma_1}$
- ▶ $z(I)$ is the length of an optimal tour of the instance I

The reference instance:

- ▶ 5 cities; the tour $(1, 3, 5, 2, 4)$ has length 18

Counting the tours

Introduction: Counting the tours

- ▶ How many tours does a TSP instance have? In this section we count them exactly
- ▶ The count tells us whether listing all the tours is a realistic way to solve the TSP
- ▶ On the reference instance the enumeration finds the optimal tour; with 20 cities it is already hopeless

Proposition 1

A TSP instance with $n \geq 3$ cities has

$$\frac{n!}{2n} = \frac{(n-1)!}{2}$$

distinct tours.

Where the count comes from

- ▶ There are $n!$ permutations of the cities.
- ▶ Each tour is described by n permutations, one for every possible starting city, so there are $n!/n = (n - 1)!$ cyclic orderings of the cities.
- ▶ The distances are symmetric, so the tour travelled backwards has the same length and is not a new solution: every tour is counted twice, and we divide by 2.

Example: enumeration of the tours

The reference instance has $\frac{(5-1)!}{2} = 12$ distinct tours, which can be enumerated:

tour	L	tour	L	tour	L
(1, 3, 5, 2, 4)	18	(1, 3, 4, 2, 5)	22	(1, 2, 3, 5, 4)	25
(1, 3, 2, 5, 4)	19	(1, 4, 3, 2, 5)	23	(1, 3, 2, 4, 5)	25
(1, 2, 5, 3, 4)	22	(1, 2, 4, 5, 3)	24	(1, 4, 2, 3, 5)	25
(1, 2, 5, 4, 3)	22			(1, 2, 4, 3, 5)	28
				(1, 2, 3, 4, 5)	29

Hence the tour $\sigma = (1, 3, 5, 2, 4)$ is the unique optimal tour, and $z(I) = 18$.

The number of tours explodes

- ▶ The number of tours grows faster than any exponential function:

n	5	10	15	20
$\frac{(n-1)!}{2}$	12	181 440	$\approx 4.4 \cdot 10^{10}$	$\approx 6.1 \cdot 10^{16}$

- ▶ A computer that evaluates 10^9 tours per second needs about $6.1 \cdot 10^7$ seconds, that is about two years, to list the tours of 20 cities.
- ▶ Enumeration is therefore useful only as a check on tiny instances, and the algorithms of this course must do better.

Summary

Counting the tours:

- ▶ a symmetric instance with n cities has $(n - 1)!/2$ tours
- ▶ $n!$ permutations, divided by n (starting city) and by 2 (direction)

Enumeration:

- ▶ on the reference instance: 12 tours, unique optimal tour $(1, 3, 5, 2, 4)$, $z(I) = 18$
- ▶ with 20 cities: about $6.1 \cdot 10^{16}$ tours, about two years of computation

A first tour by hand

Introduction: A first tour by hand

- ▶ Enumeration is hopeless beyond tiny instances: we need a fast way to build a tour
- ▶ We see the simplest one, the nearest neighbour rule, and we apply it by hand to the reference instance
- ▶ We compare the tour with the optimal one, and see why a tour alone is not enough

The nearest neighbour rule

- ▶ The simplest way to build a tour is the **nearest neighbour rule**:
 - ▶ start from a city;
 - ▶ as long as there are unvisited cities, move to the closest unvisited one;
 - ▶ at the end, return to the starting city.
- ▶ In case of a tie we take the city with the smallest index.

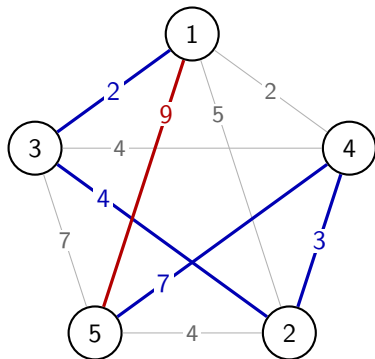
Example: nearest neighbour from city 1 (1/2)

On the reference instance, starting from city 1:

1. from 1 the closest cities are 3 and 4, both at distance 2: we go to 3;
2. from 3 the unvisited cities 2 and 4 are both at distance 4: we go to 2;
3. from 2 the closest unvisited city is 4, at distance 3; then only 5 is left, at distance $d_{45} = 7$;
4. we return from 5 to 1 at distance $d_{51} = 9$.

The tour is $\sigma = (1, 3, 2, 4, 5)$, of length $2 + 4 + 3 + 7 + 9 = 25$.

Example: nearest neighbour from city 1 (2/2)



- ▶ the bold edges form the tour $\sigma = (1, 3, 2, 4, 5)$
- ▶ the cities are in the positions of the drawing of the reference instance
- ▶ the last edge $\{5, 1\}$, in red, is forced

Its length is

$$L(\sigma) = d_{13} + d_{32} + d_{24} + d_{45} + d_{51} = 2 + 4 + 3 + 7 + 9 = 25.$$

How good is the nearest neighbour tour?

- ▶ The last edges of the nearest neighbour tour are forced, and may be very long: here the final edge $\{5, 1\}$ costs 9.
- ▶ The tour of length 25 is worse than the optimal one, of length 18, by 7.
- ▶ From the nearest neighbour tour alone we cannot tell how far we are from the optimum: we need a lower bound.

Summary

The nearest neighbour rule:

- ▶ always move to the closest unvisited city (ties: smallest index), then return to the start
- ▶ it builds a tour quickly, with no guarantee

On the reference instance:

- ▶ from city 1: the tour (1, 3, 2, 4, 5) of length 25, worse than the optimum 18 by 7
- ▶ to know how far a tour is from the optimum we need a lower bound

A combinatorial lower bound

Introduction: A combinatorial lower bound

- ▶ A tour gives an upper bound on $z(I)$; to know how far it is from the optimum we need a lower bound
- ▶ We build a simple lower bound from the two shortest edges of every city
- ▶ On the reference instance we use it to measure the gap of the nearest neighbour tour

- For a city i , let $a_i \leq b_i$ be the two smallest distances from i to the other cities.

Observation 1

For every symmetric TSP instance, the optimal length satisfies

$$z(I) \geq \left\lceil \frac{1}{2} \sum_{i \in \mathcal{V}} (a_i + b_i) \right\rceil.$$

Why the bound holds

- ▶ In a tour, every city is entered by one edge and left by another one, so it has exactly two edges of the tour, each at least as long as a_i and b_i respectively.
- ▶ Summing over all the cities, every edge of the tour is counted twice, once for each of its end cities: hence $2L(\sigma) \geq \sum_i (a_i + b_i)$ for every tour σ .
- ▶ The distances are integer, so is $L(\sigma)$, and we can round the bound up.

Example: the bound on the reference instance (1/2)

For the reference instance the two smallest distances of each city are:

city i	1	2	3	4	5	sum
a_i	2	3	2	2	4	
b_i	2	4	4	3	7	
$a_i + b_i$	4	7	6	5	11	33

Hence

$$z(I) \geq \lceil 33/2 \rceil = 17.$$

Example: the bound on the reference instance (2/2)

- ▶ Together with the nearest neighbour tour we obtain

$$17 \leq z(I) \leq 25,$$

an absolute gap of 8, that is a relative gap of $\frac{8}{17} \approx 47.1\%$.

- ▶ The tour (1, 3, 5, 2, 4) of length 18 closes most of it, and only the enumeration of the 12 tours proves that 18 is optimal.

Summary

The two-shortest-edges bound:

- ▶ $a_i \leq b_i$ are the two smallest distances of city i
- ▶ every city has two edges of the tour, and every edge is counted twice: $z(I) \geq \lceil \frac{1}{2} \sum_{i \in \mathcal{V}} (a_i + b_i) \rceil$

On the reference instance:

- ▶ bound 17; with the nearest neighbour tour, $17 \leq z(I) \leq 25$, a gap of 8
- ▶ the optimal length 18 is proved only by enumeration

Variants: metric and asymmetric TSP

Introduction: Variants of the TSP

- ▶ Two important variants of the TSP are obtained by changing the properties of the distances
- ▶ In the metric TSP the distances satisfy the triangle inequality: it is the case where the heuristic algorithms have guarantees
- ▶ In the asymmetric TSP the distance from i to j may differ from the one from j to i : the direction of a tour matters

The metric travelling salesman problem

Definition 2: Metric TSP

A TSP instance is **metric** if its distances satisfy the **triangle inequality**

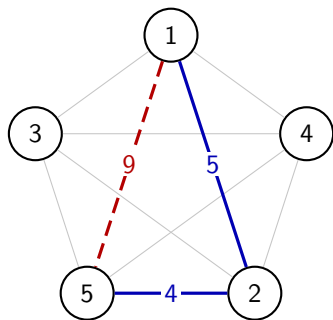
$$d_{ij} \leq d_{ik} + d_{kj} \quad \text{for all distinct } i, j, k \in \mathcal{V}.$$

The **metric travelling salesman problem** (metric TSP) is the TSP restricted to metric instances.

- ▶ The triangle inequality says that going directly from i to j is never longer than going through a third city k .

The reference instance is metric

- ▶ All the $5 \cdot 4 \cdot 3 = 60$ triangle inequalities hold, some of them with equality, e.g., $d_{15} = 9 = d_{12} + d_{25} = 5 + 4$.
- ▶ In the drawing, the direct edge $\{1, 5\}$ (dashed) is as long as the path from 1 to 5 through city 2.



$$d_{15} = 9 \leq d_{12} + d_{25} = 5 + 4$$

Where metric instances come from

- ▶ Instances obtained from points in the plane, with d_{ij} equal to the Euclidean distance between the points, are metric (when the distances are not rounded).
- ▶ Several heuristic algorithms for the TSP behave well, and some of them have a performance guarantee, only on metric instances.

The asymmetric travelling salesman problem

Definition 3: Asymmetric travelling salesman problem

Given:

1. a set $\mathcal{V} = \{1, 2, \dots, n\}$ of n cities,
2. for each ordered pair of distinct cities $i, j \in \mathcal{V}$, a distance $d_{ij} \in \mathbb{Z}_{\geq 1}$ from city i to city j , where d_{ij} may differ from d_{ji} ,

the **asymmetric travelling salesman problem (ATSP)** asks to find a closed route that starts from a city, visits every other city exactly once and returns to the starting city, so that the total distance travelled is minimized.

The ATSP and the TSP

- ▶ Asymmetric distances arise, for example, when travelling along one-way streets or uphill and downhill.
- ▶ The definition of tour and of its length $L(\sigma)$ is unchanged, but the direction now matters: a tour and the same tour travelled backwards are different solutions.
- ▶ The TSP is the special case of the ATSP in which $d_{ij} = d_{ji}$ for all i, j .

Counting the tours of the ATSP

- ▶ An ATSP instance has $\frac{n!}{n} = (n-1)!$ distinct tours, twice the count of the proposition for the TSP:
 - ▶ the starting city still does not change the tour;
 - ▶ but the direction does, so we do not divide by 2.
- ▶ The reference instance, read as an ATSP instance, has $4! = 24$ tours instead of 12.

Summary

Metric TSP:

- ▶ the distances satisfy the triangle inequality $d_{ij} \leq d_{ik} + d_{kj}$
- ▶ the reference instance is metric; Euclidean distances (not rounded) are metric
- ▶ it is the case where heuristic algorithms can have a performance guarantee

Asymmetric TSP:

- ▶ d_{ij} may differ from d_{ji} : the direction of a tour matters
- ▶ $(n-1)!$ tours instead of $(n-1)!/2$; the TSP is a special case

Remarks

Introduction: Remarks

- ▶ We close the chapter with a few remarks on the state of the art
- ▶ How large are the TSP instances that are solved to optimality today?
- ▶ What does complexity say about the TSP, and what can an approximation algorithm guarantee?

Record instances

- ▶ The largest instance solved to optimality, pla85900, has 85 900 points from a chip design problem.
- ▶ It was solved in 2005–2006 with the Concorde code; the optimal tour, of length 142 382 641, was certified in 2009 by Applegate, Bixby, Chvátal, Cook, Espinoza, Goycoolea and Helsgaun.
- ▶ On the road maps of real countries, the record of 2025 is the instance korea81998, with 81 998 bars of South Korea.

Complexity and approximation

- ▶ The TSP is strongly NP-hard (chapter [«Combinatorial optimization problems»](#)).
- ▶ Christofides (1976) gave an approximation algorithm with ratio $\frac{3}{2}$ for the metric TSP.

Summary

Exact solution:

- ▶ record instances with tens of thousands of points: pla85900 and korea81998

Complexity and approximation:

- ▶ the TSP is strongly NP-hard
- ▶ for the metric TSP, Christofides' algorithm has ratio $\frac{3}{2}$

Typical mistakes

Introduction: Typical mistakes

- ▶ We collect the mistakes that are made most often with the TSP
- ▶ Each of them comes from a definition or a result of this chapter
- ▶ Check them before solving the exercises

Typical mistakes (1/3)

Observation 2: Counting $n!$ tours

The starting city does not change the tour, and in the symmetric case neither does the direction: the number of tours is $(n - 1)!/2$.

Observation 3: Forgetting the return edge

The length of a tour includes the last distance $d_{\sigma_n \sigma_1}$, from the last city back to the first one.

Typical mistakes (2/3)

Observation 4: Taking two cycles for a tour

Two disjoint cycles that together visit every city once are not a tour: a tour is a single cycle through all the cities.

Observation 5: Reading the bound as a tour

The value 17 of the two-shortest-edges bound is not the length of a tour; it only says that no tour is shorter.

Typical mistakes (3/3)

Observation 6: Using the triangle inequality without checking it

The guarantees of the heuristic algorithms hold only for metric instances.

Summary

Tours:

- ▶ $(n - 1)!/2$ tours, not $n!$; the length includes the return edge
- ▶ a tour is a single cycle through all the cities

Bounds and guarantees:

- ▶ a lower bound is not the length of a tour
- ▶ check the triangle inequality before using a guarantee for metric instances

In summary

Introduction: In summary

- ▶ We collect the key ideas of the chapter on routing problems
- ▶ The dashboard gathers the bounds found on the reference instance

Summary (1/2)

- ▶ The TSP asks for a shortest closed route through all the cities; a solution is a cyclic ordering of the cities, called a tour.
- ▶ A symmetric instance with n cities has $(n - 1)!/2$ tours, an asymmetric one $(n - 1)!$; listing them is possible only for very small n .
- ▶ The nearest neighbour rule gives a tour quickly, with no guarantee; the two shortest edges of every city give a lower bound on the optimal length.
- ▶ The metric TSP, with the triangle inequality, is the case where approximation guarantees exist.

Summary (2/2)

Dashboard of the TSP instance (minimization: the lower bound is a dual bound, the upper bound is the length of a tour):

Chapter	Lower bound	Upper bound	From
This chapter	17	25	two shortest edges; nearest neighbour from city 1
This chapter, enumeration	18	18	the 12 tours