SA for TSP

Problem:

- Given n cities,
- Find a complete tour with minimal length.

Search space is **BIG**: for 30 cities there are

$$30! \quad 10^{32}$$

possible tours.

SA for TSP

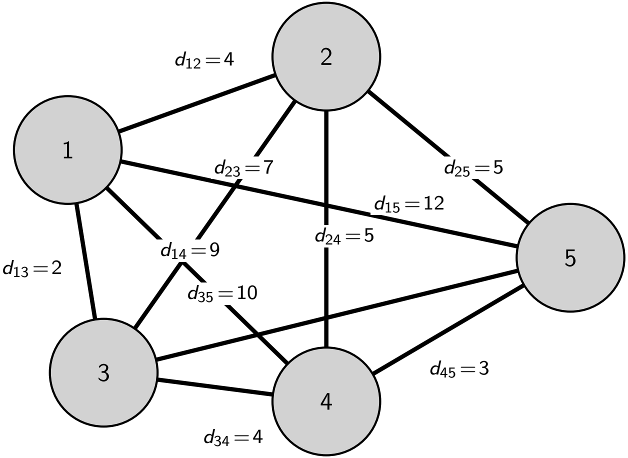
The solution is a permutation of n cities:

$$\text{Sol} = [1,\ 2,\ 3,\ \ldots,\ n]$$

Several choices exist for generating the required neighbourhood.

- The new solution could be generated by performing a predetermined number of swaps on the current solution,
- Some adaptive implementations decrease the number of cities to swap during the run. Thus, reducing the neighbourhood size as the search progresses.

SA for TSP – Example



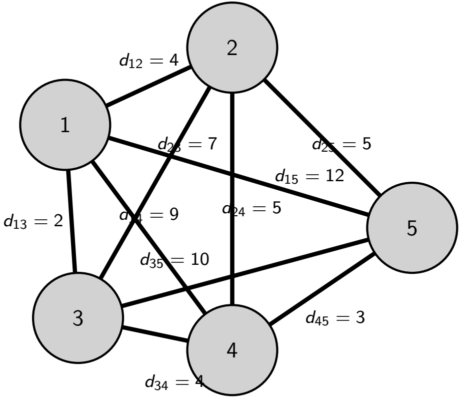
SA for TSP – Example

Start with a random solution :

$$\textbf{Sol} = [\,1,\ 3,\ 4,\ 2,\ 5\,]$$

The total distance would be:

$$total\_dist = \sum_{i=1}^{n-1} d_{\mathrm{Sol}_i,\mathrm{Sol}_{i+1}} + \underbrace{d_{\mathrm{Sol}_n,\mathrm{Sol}_1}}_{\text{For a closed tour}}$$



SA for TSP – Example

The tour length of the initial solution is:

$$\text{Sol} = [\,1, 3, 4, 2, 5\,]$$

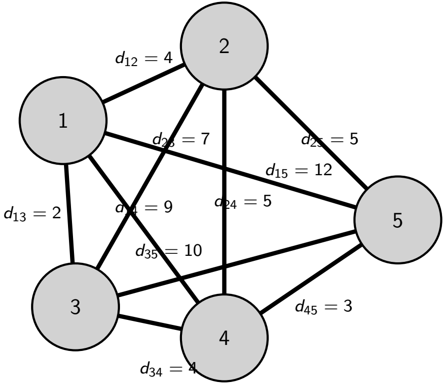
$$total\_dist = 2 + 4 + 5 + 5 + 12 = 28$$

To generate a candidate solution, select two random cities and swap them:

$$\text{Sol} = [\,1,\ \textcircled{2},\ 4,\ \textcircled{3},\ 5\,]$$

The tour length of the candidate solution is:

$$total\_dist = 4 + 5 + 4 + 10 + 12 = 35$$

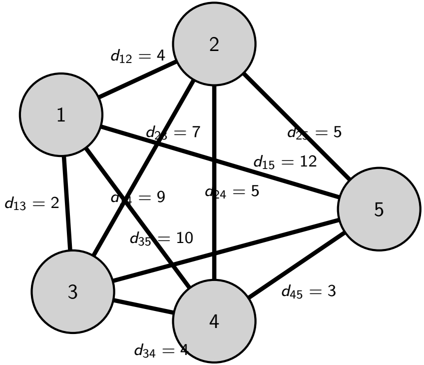


SA for TSP - Example

Since the new solution has a longer tour length, it will be conditionally accepted according to a probability of (at higher temperatures, there ' s a higher probability of acceptance): $P = e^{-7/t}$

- Assuming the new solution was not accepted, we generate a different one starting from the initial solution:

$$Sol = [\,1,\ 3,\ 4,\ 5,\ 2\,]$$

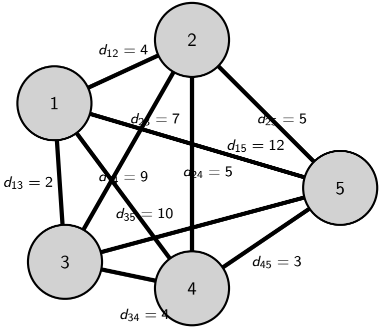


SA for TSP - Example

The tour length of the candidate solution [1, 3, 4, 2, 5] is:

$$total\_dist = 2 + 4 + 3 + 5 + 4 = 18$$

Since this solution has a shorter tour length, it will get accepted.
The search continues ...



SA for berlin52 (TSPLIB) — Experiment Setup

The SA was applied for the `berlin52` TSP instance by having:

- $T_{\text{initial}} = 10000$,
- $T_{\text{Final}} = 0.1$,
- $\alpha = 0.85$,
- Using 1000 iteration for each T
- Using different number of swaps for obtaining the neighbouring solution (1, 5, 10 and 15),

Note. The number of swaps controls the *neighborhood radius*:

- 1 swap $\Rightarrow$ small local perturbation (high acceptance, slow exploration),
- 15 swaps $\Rightarrow$ larger perturbation (lower acceptance, stronger exploration).

```
NAME: berlin52
TYPE: TSP
COMMENT: 52 locations in Berlin (Groetschel)
DIMENSION: 52
EDGE_WEIGHT_TYPE: EUC_2D
NODE_COORD_SECTION
1 565.0 575.0
2 25.0 185.0
3 345.0 750.0
4 945.0 685.0
5 845.0 655.0
6 880.0 660.0
7 25.0 230.0
8 525.0 1000.0
9 580.0 1175.0
10 650.0 1130.0
11 1605.0 620.0
12 1220.0 580.0
13 1465.0 200.0
14 1530.0 5.0
15 845.0 680.0
16 725.0 370.0
17 145.0 665.0
18 415.0 635.0
19 510.0 875.0
20 560.0 365.0
21 300.0 465.0
22 520.0 585.0
23 480.0 415.0
24 835.0 625.0
25 975.0 580.0
26 1215.0 245.0
27 1320.0 315.0
28 1250.0 400.0
29 660.0 180.0
30 410.0 250.0
31 420.0 555.0
32 575.0 665.0
33 1150.0 1160.0
34 700.0 580.0
35 685.0 595.0
36 685.0 610.0
37 770.0 610.0
38 795.0 645.0
39 720.0 635.0
40 760.0 650.0
41 475.0 960.0
42 95.0 260.0
43 875.0 920.0
44 700.0 500.0
45 555.0 815.0
46 830.0 485.0
47 1170.0 65.0
48 830.0 610.0
49 605.0 625.0
50 595.0 360.0
51 1340.0 725.0
52 1740.0 245.0
EOF
```

SA for berlin52 — Effect of Neighborhood Size

Number of swaps	Tour length
1	8861
5	13180
10	17124
15	18900

Observation.

- Increasing the number of swaps increases the neighborhood radius.
- Larger neighborhoods introduce stronger perturbations, often producing longer tours when the temperature schedule is unchanged.
- This highlights the trade-off between *exploration strength* and *solution quality*.

Probability of Acceptance

The Boltzmann-based acceptance probability takes significant computational time ($\sim 1/3$ of the SA computations).

The idea of using a lookup table where the exponential calculation are done once for a range of values for change in c and t has been suggested.

Other non-exponential probability formulas have been suggested, for example Johnson et. al. [8] suggested

$$P(\delta c) = 1 - \frac{\delta c}{t}.$$

Why Adaptive Simulated Annealing?

Classical SA limitation

Standard SA relies on a **fixed cooling schedule**:

$$T_0 \rightarrow T_1 \rightarrow \dots \rightarrow 0$$

which assumes prior knowledge of the problem landscape.

- Too fast cooling $\Rightarrow$ premature convergence
- Too slow cooling $\Rightarrow$ wasted computation
- One schedule rarely fits all phases of the search

Key idea

Adaptation adjusts SA parameters online using feedback from the search process itself.

What Can Be Adapted in Simulated Annealing?

Primary targets

- Temperature $T(t)$
- Cooling rate $\alpha(t)$
- Neighborhood step size

Secondary targets

- Acceptance ratio
- Reheating triggers
- Restart thresholds

Unifying principle

Adaptation aims to **maintain effective exploration** while gradually enforcing exploitation.

Reheating and Stagnation-Triggered Adaptation

Stagnation detection

Define:

$$g(t) = \text{iterations since last improvement}$$

Reheating rule (example)

$$\text{if } g(t) > G_{\max} \quad \Rightarrow \quad T(t+1) = \kappa\, T_0$$

- Escapes deep local minima
- Restores probabilistic uphill moves
- Often combined with neighborhood expansion

Interpretation

Reheating is a form of **explicit diversification**.