The Traveling Salesman Problem

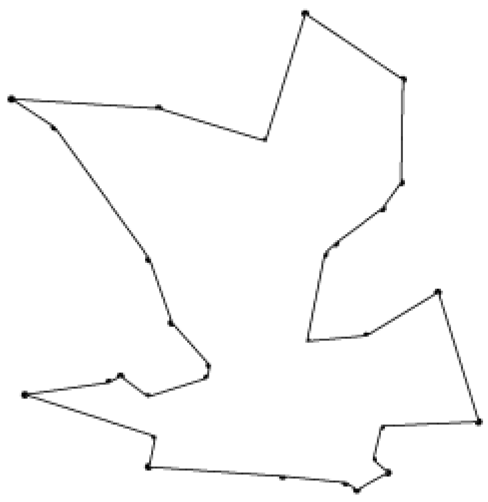
Aparajithan Venkateswaran

APPM 5120

University of Colorado Boulder

The Problem

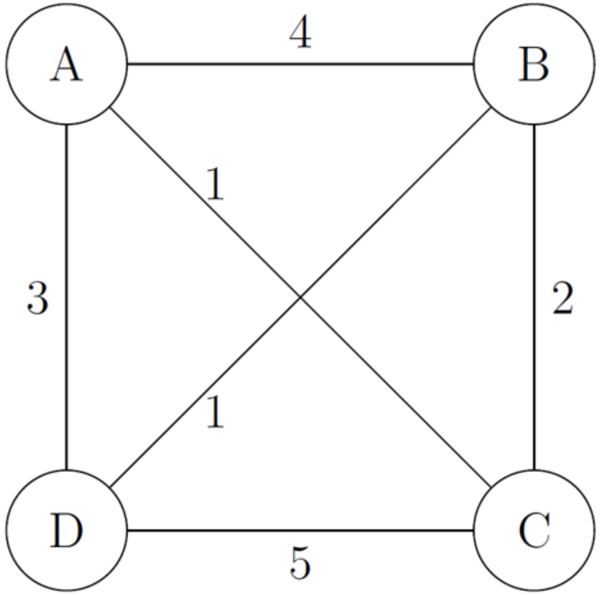
Given a list of different cities and the distance between pairs of cities, what is the shortest route that visits all cities exactly once and returns to the origin city?



Graphs are awesome!

Basic Graph Theory

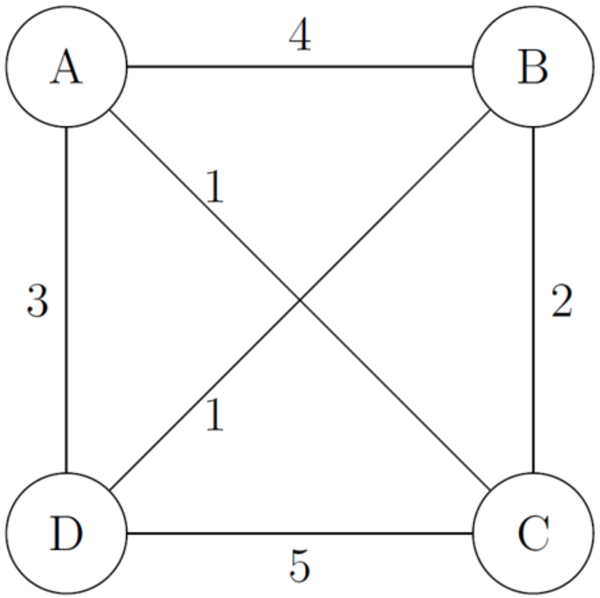
Think dots and lines connecting them



Basic Graph Theory

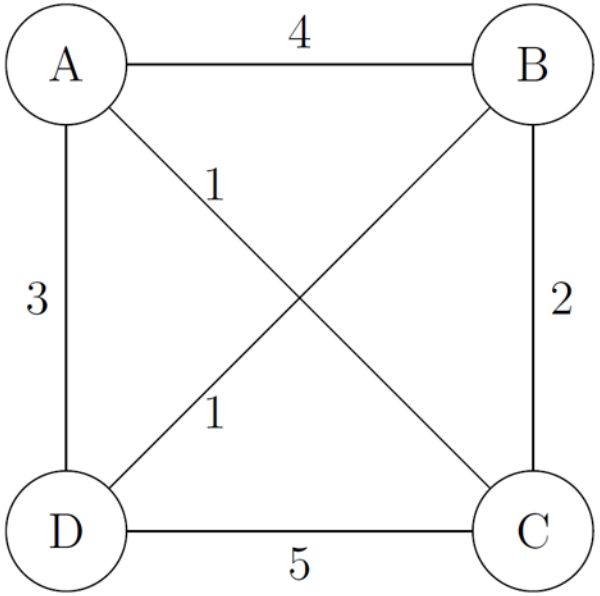
$$G = (V, E)$$
$$V = \text{set of vertices}$$
$$E = \text{set of edges}$$
$$E \subseteq V \times V$$

Basic Graph Theory



$$G = (V, E)$$
$$V = \text{set of vertices}$$
$$E = \text{set of edges}$$

Basic Graph Theory



$$G = (V, E)$$
$$V = \{A, B, C, D\}$$
$$E = \{(A, B), (A, C), (A, D), (B, C), (B, D), (C, D)\}$$

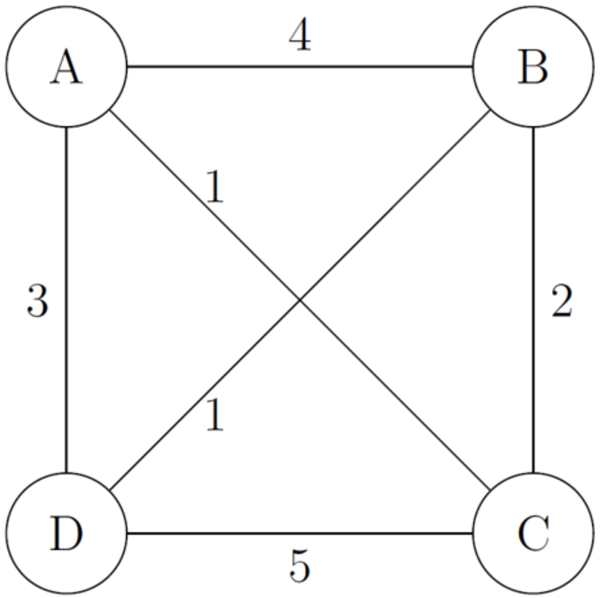
Basic Graph Theory

We can also have a weight function associated with edges

- Could represent interactions between different species
- Correlations between genes
- The cost of travelling between cities

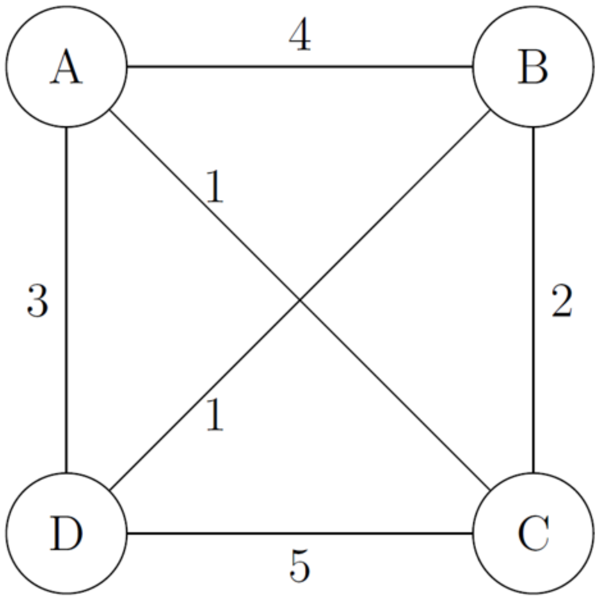
$$w : e \to \mathbb{R}_0^+$$
$$e \in E$$

Basic Graph Theory



$$w(A, B) =$$
$$w(D, C) =$$

Basic Graph Theory

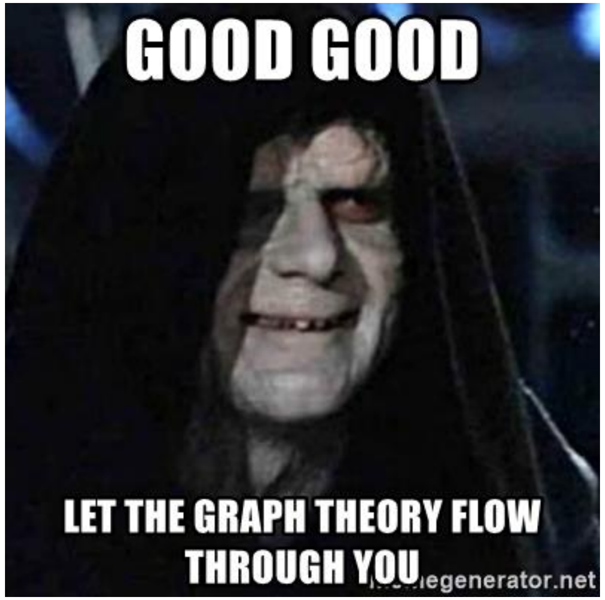


$$w(A, B) = 4$$
$$w(D, C) = 5$$

Basic Graph Theory

Important takeaways:

- Graphs are an amazing way to represent correlated data
- We can represent our network of cities as a graph



Formulating the Problem

Formulating the Problem

Let us define a new variable

$$x_{ij} = \begin{cases} 1 \text{ if the shortest path goes from city } i \text{ to city } j \\ 0 \text{ otherwise} \end{cases}$$

So, we have…

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} w(i,j) x_{ij}$$

So, we have…

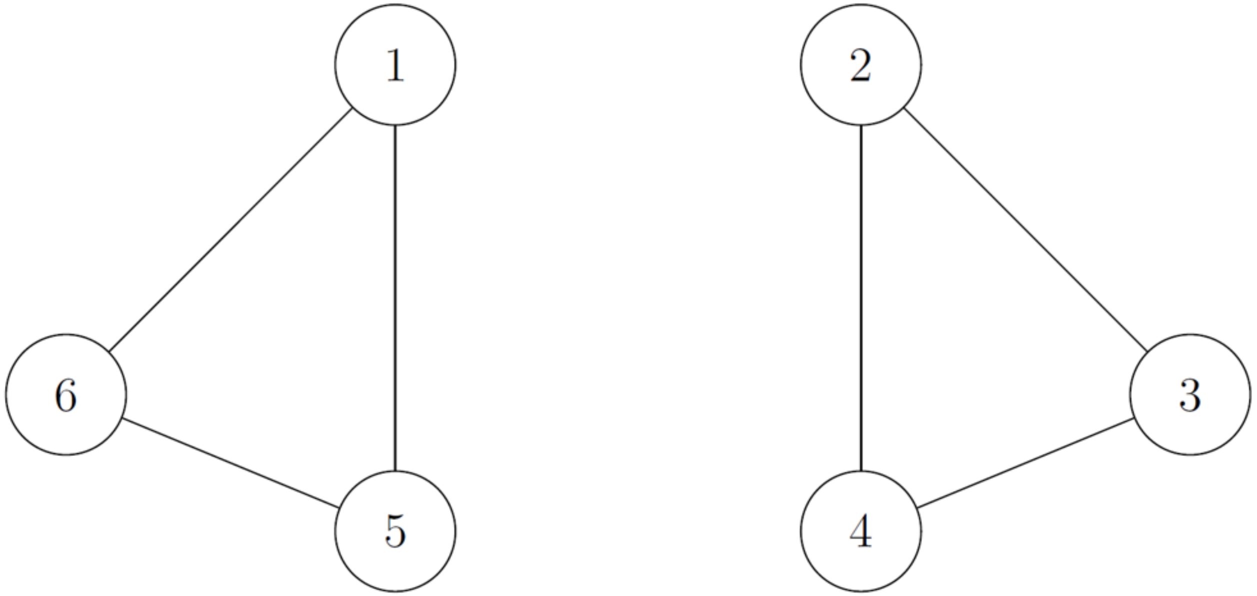
$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} w(i,j) x_{ij}$$

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1, \qquad j \in V$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1, \qquad i \in V$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V, i \neq j$$

But what if…



Solution…

Define a new “dummy” variable

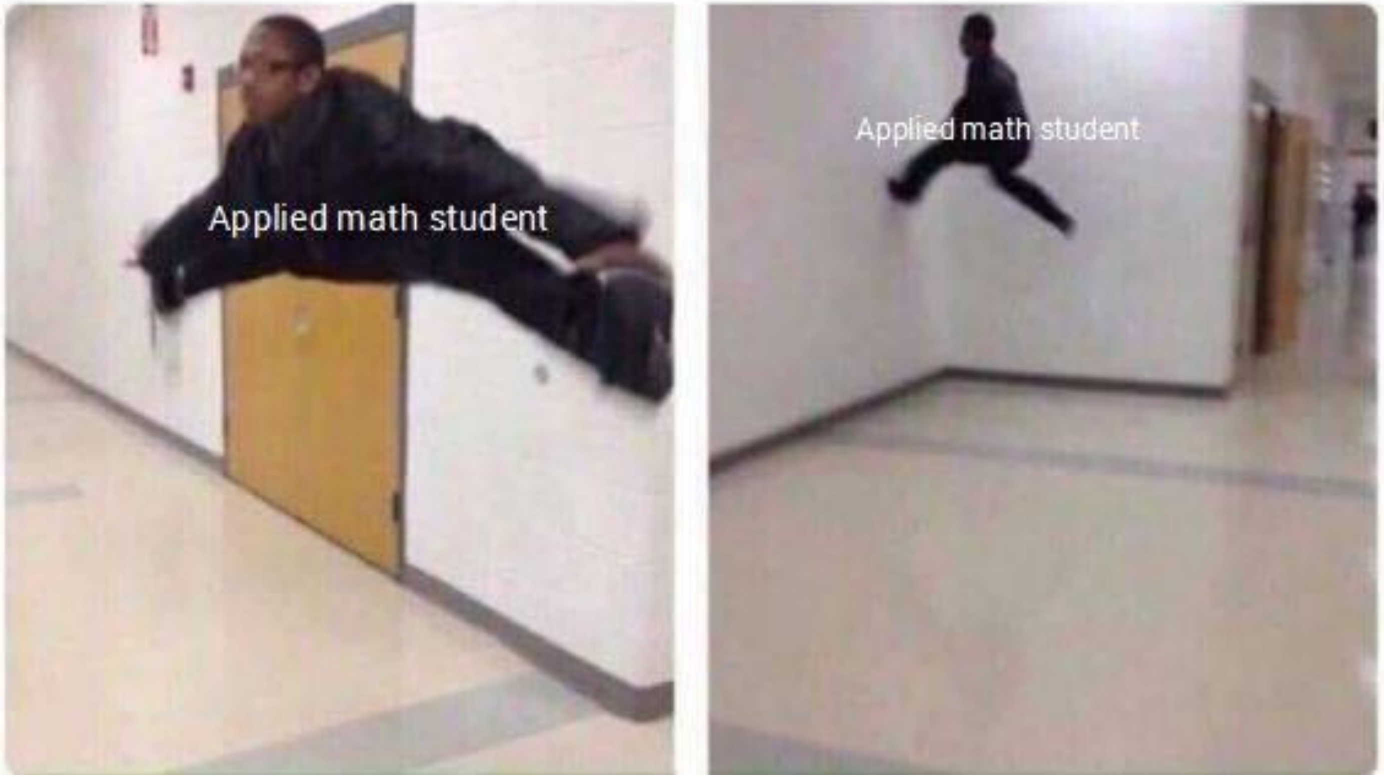
$$u_i = t$$

$$\text{city } i \text{ is visited in the } t\text{-th step}$$

Fixed!

$$
\begin{aligned}
\min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} w(i,j) x_{ij} & \\
\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1, & \qquad j \in V \\
\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1, & \qquad i \in V \\
u_i - u_j + n x_{ij} \leq n - 1, & \qquad i, j \in V,\ j \neq i,\ i, j \neq o \\
x_{ij} \in \{0, 1\} & \qquad i, j \in V, i \neq j \\
u_i \in \mathbb{Z} & \qquad i \in V
\end{aligned}
$$

the floor is **proofs**



Take my word for it, please…

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} w(i,j) x_{ij}$$

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1, \qquad j \in V$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1, \qquad i \in V$$

$$u_i - u_j + n x_{ij} \leq n - 1, \qquad i,j \in V,\ j \neq i,\ i,j \neq o$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V, i \neq j$$

$$u_i \in \mathbb{Z} \qquad i \in V$$

How do you solve it?

Computational Complexity

Enumerate all solutions and pick the one with the shortest tour?

$$O(n!)$$

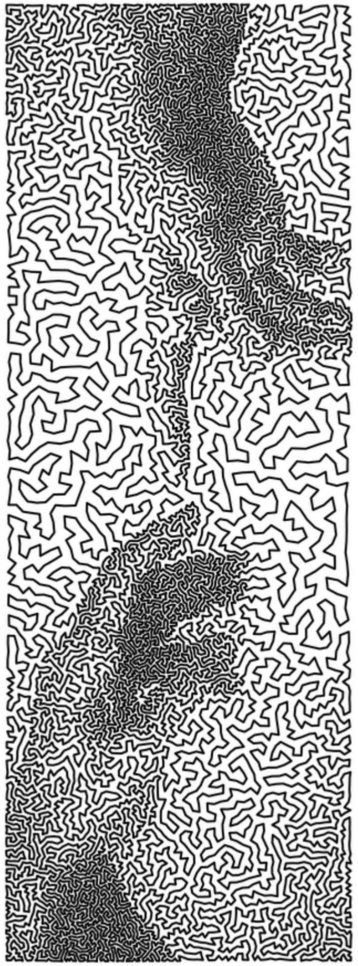
Computational Complexity

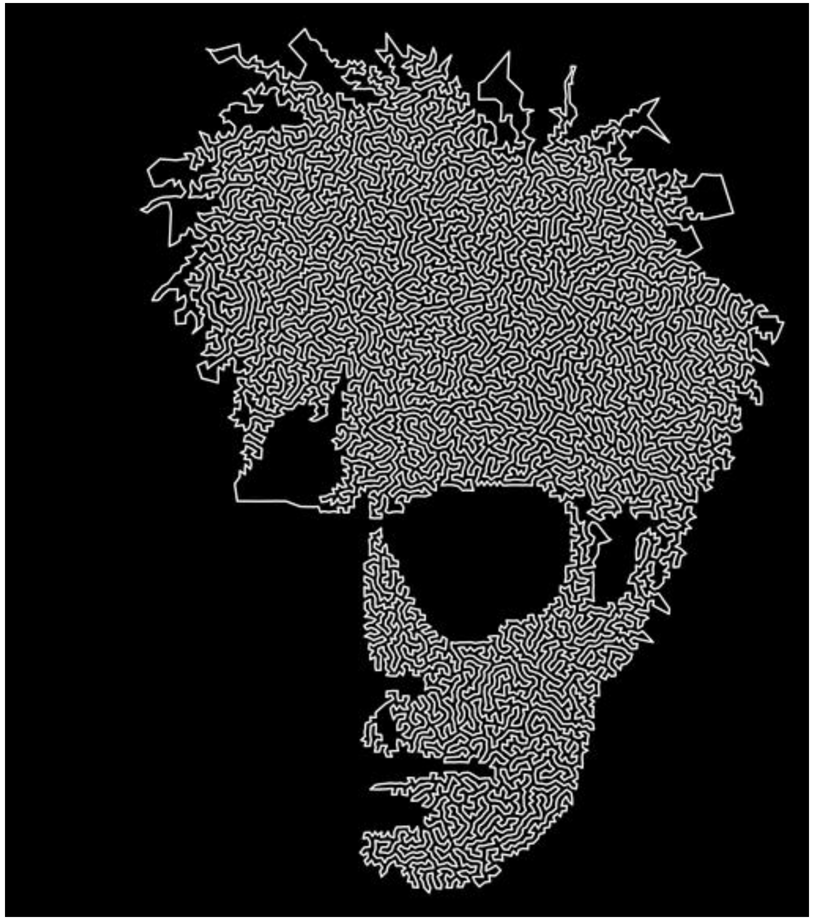
- OR techniques like Simplex Method or Interior Point Method?
- We have too many variables and constraints!
- Number of constraints grow of the order of n^2

Computational Complexity

Turns out that the problem is NP-hard!

Why bother trying to solve it?





Why bother trying to solve it?

Finds applications in various other fields

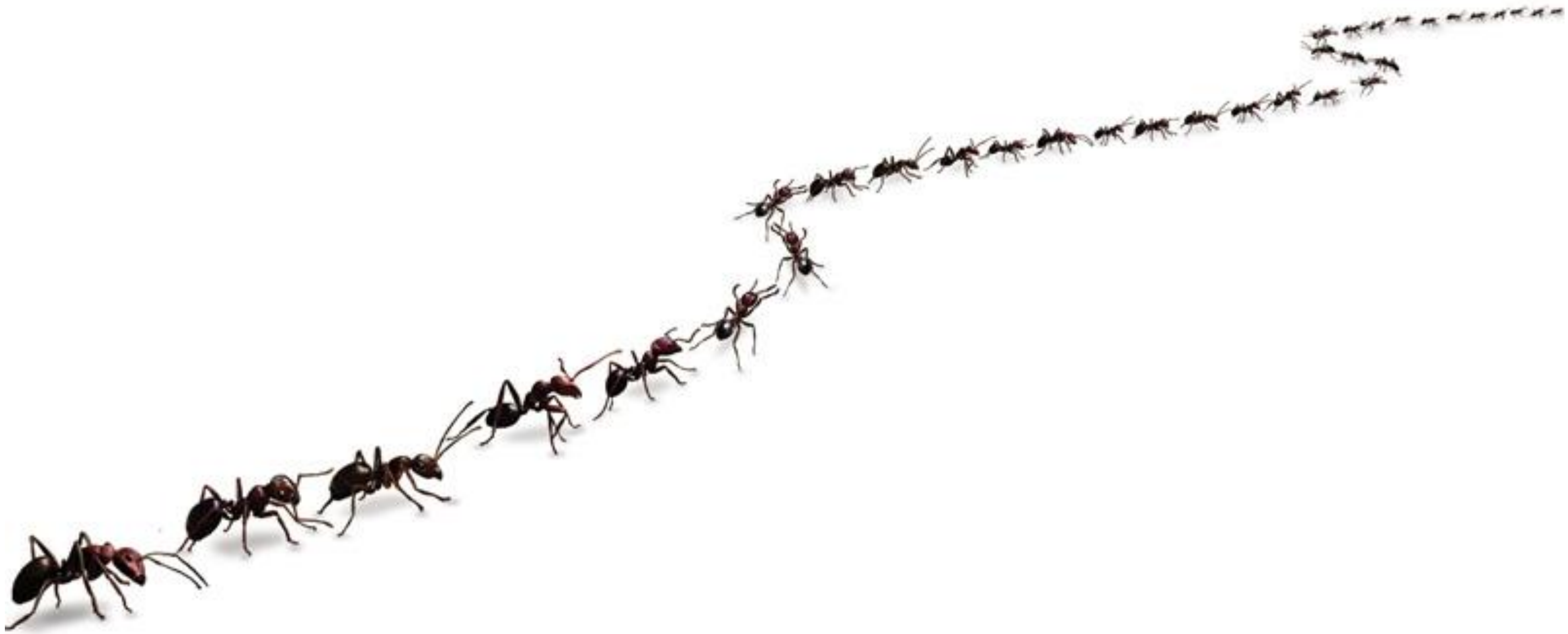
- Job scheduling
- Network routing
- Quadratic Assignment Problem
- Circuit Design
- DNA Sequencing

Heuristics

Heuristics

- A greedy approach
- Christofides' Algorithm – think Minimum Spanning Trees
- Match Twice and Stitch
- Lin-Kernighan heuristics

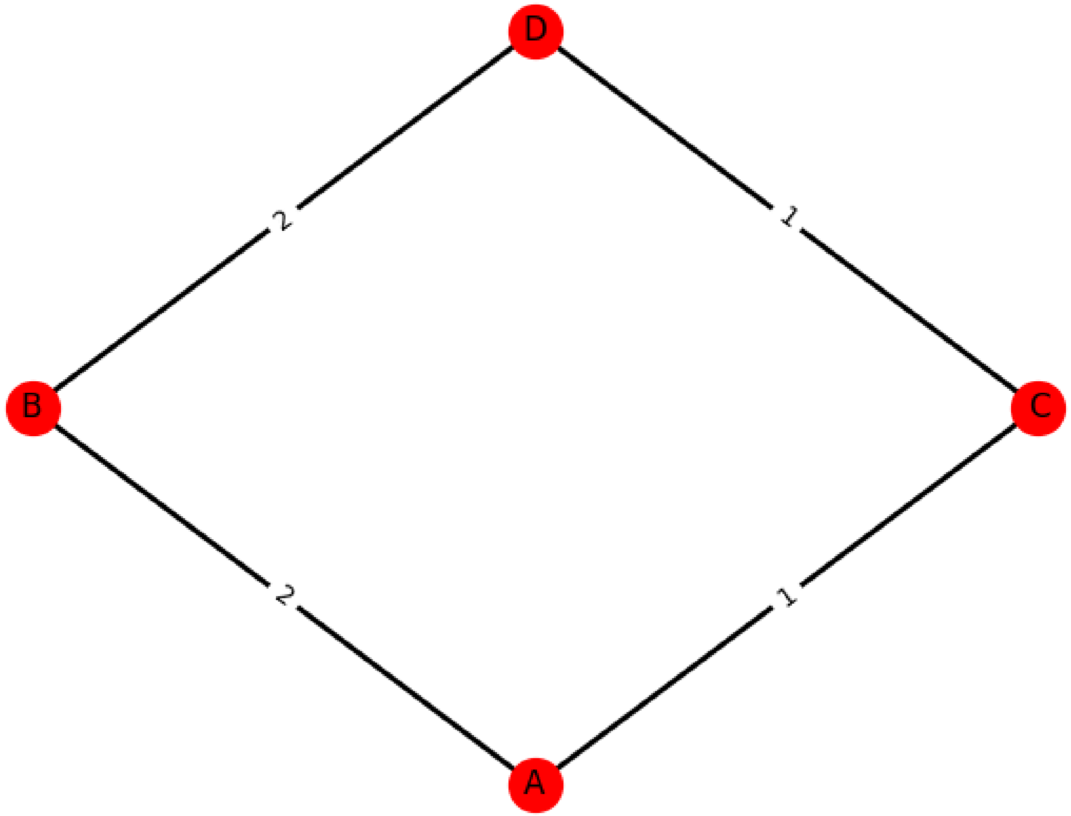
Ants are smart



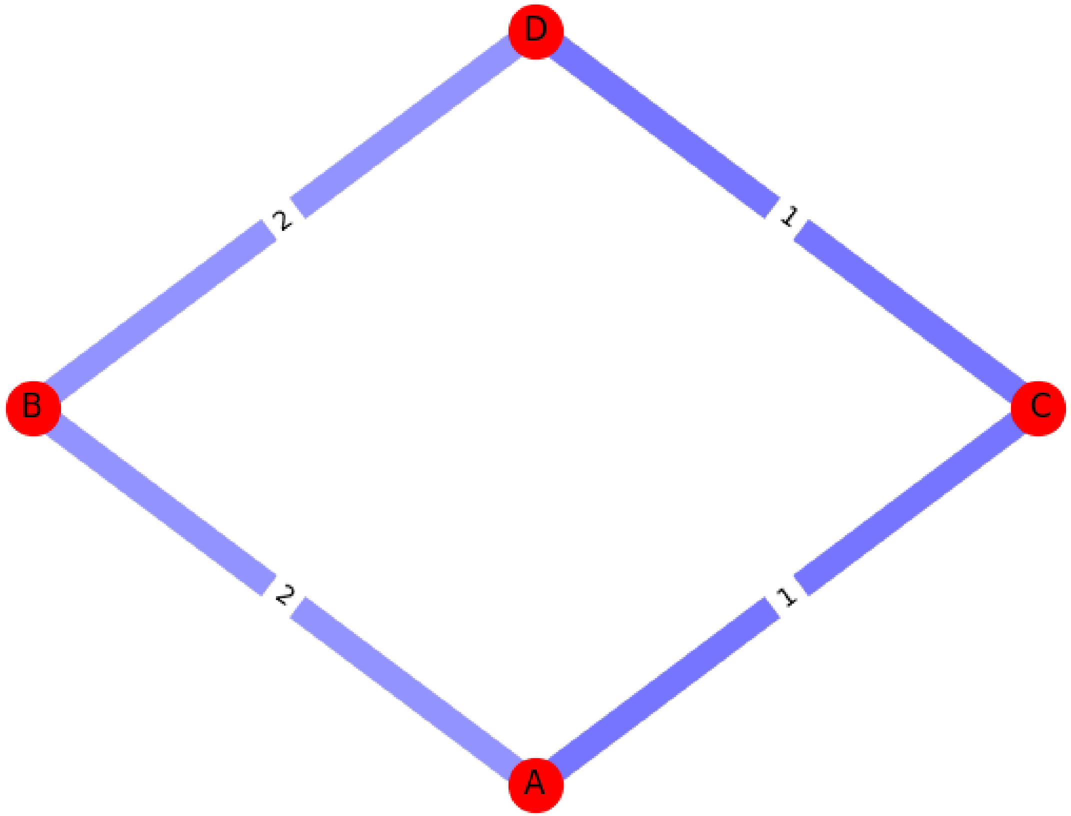
Ants are smart

- Ants walk randomly until they find food
- They go back home leaving a pheromone trail
- Another ant is more likely to follow this trail upon discovering it
- Time evaporates the pheromones
- Longer paths have lower pheromone content
- Shorter paths are preferred

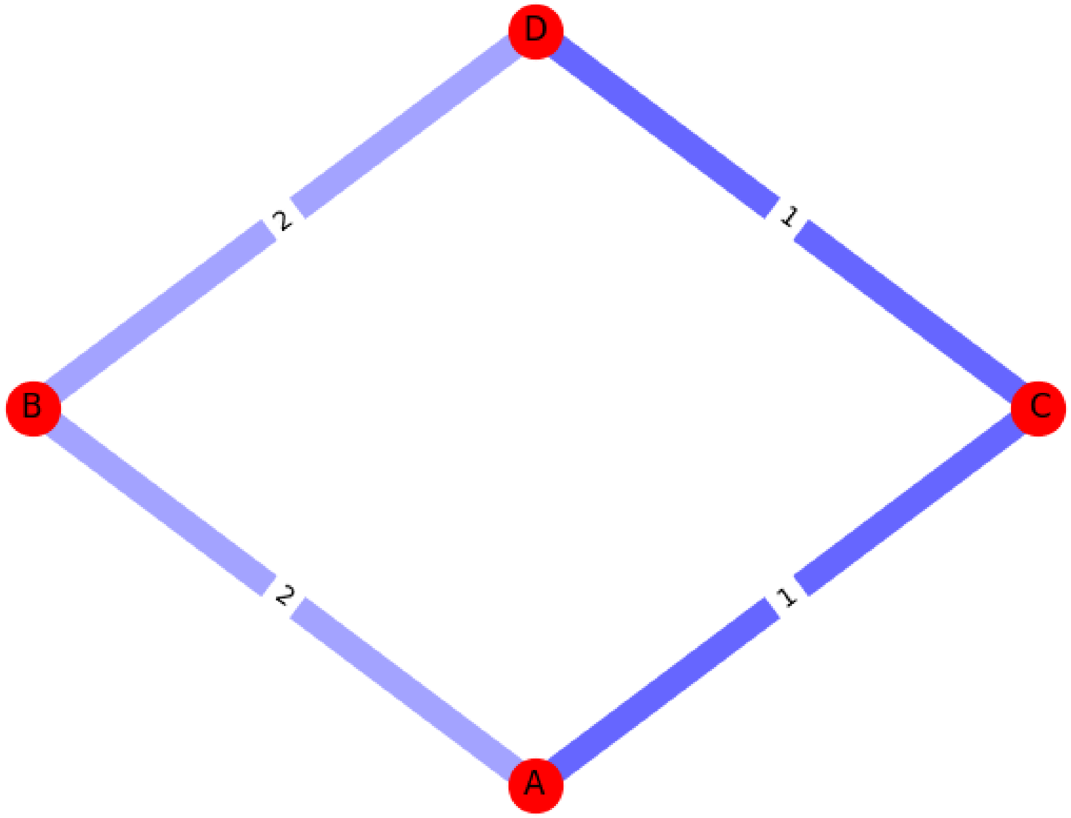
Iteration 0 (Initial State)



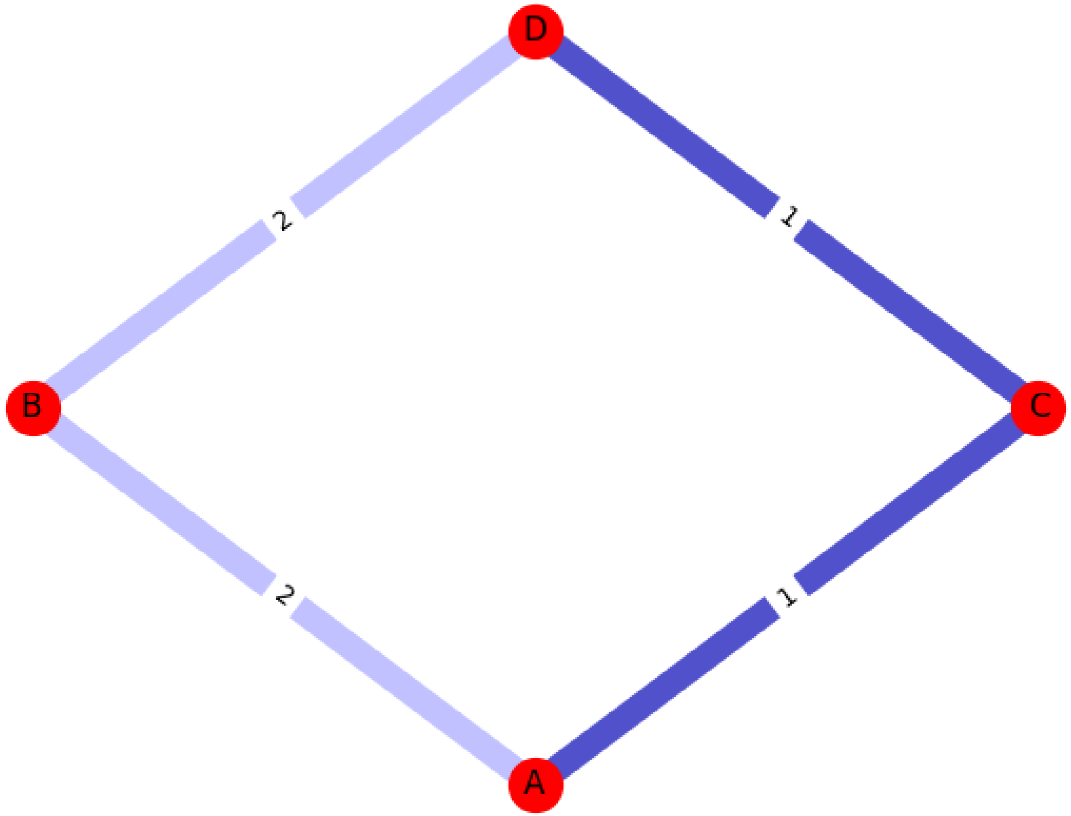
Iteration 1



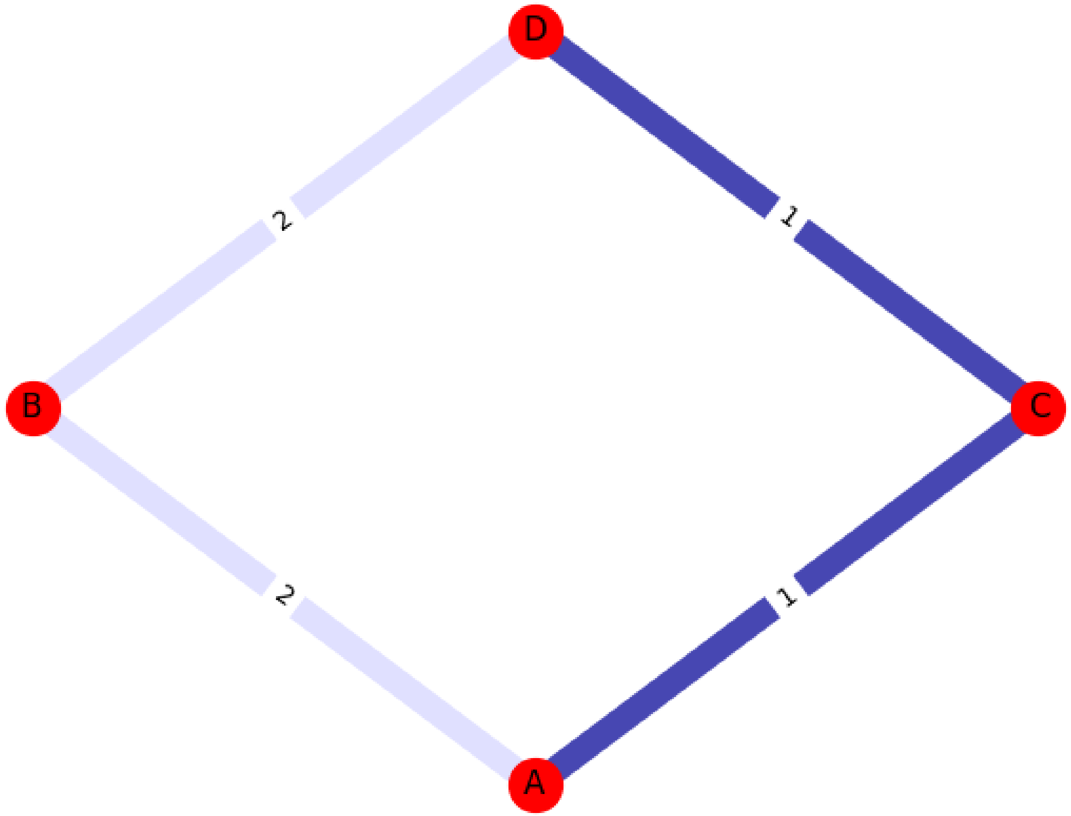
Iteration 2



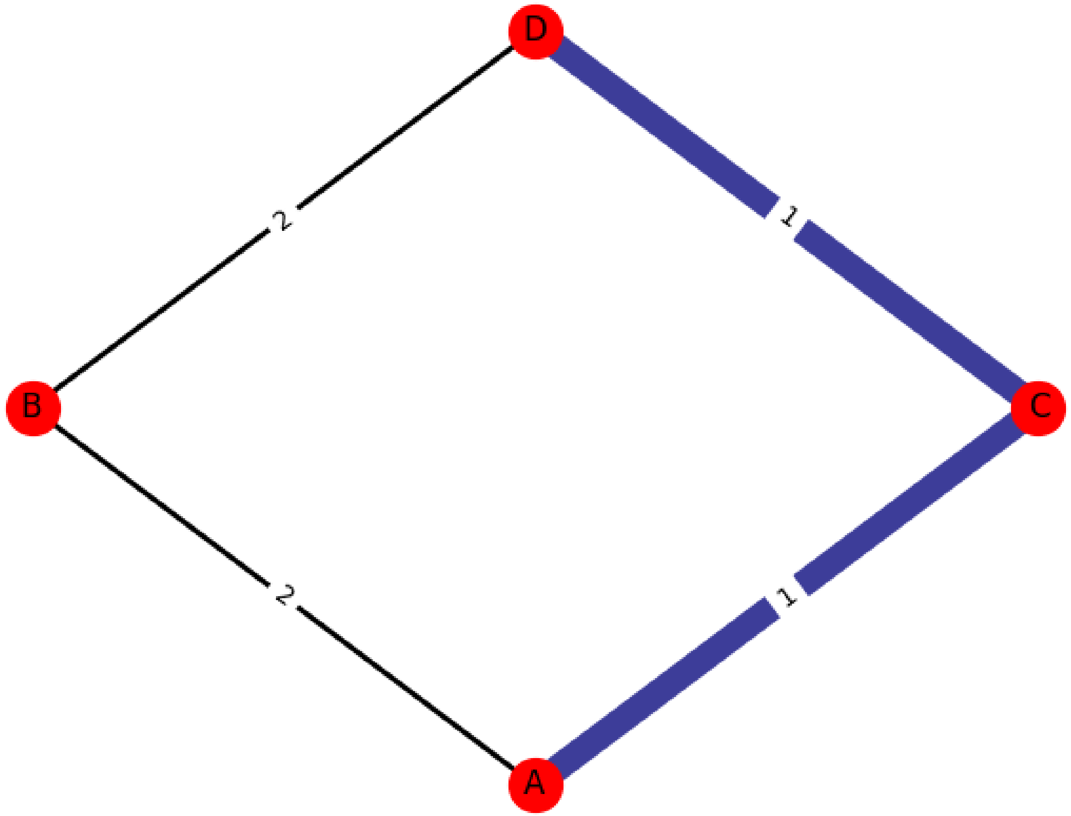
Iteration 3



Iteration 4



Iteration 5 (Finally)



Ant Colony Optimization Algorithm

- Probabilistic algorithm
- We keep track of a colony of ants
- We also keep track of the pheromone contents along the edges

```
repeat until termination {
    completeTours()         // allow ants to complete tours
    updatePheromoneMap()    // update the pheromone contents
}
```

Completing Tours

- Choice of edge depends on the amount of pheromones present

$$\tau(i,j) : \text{Amount of pheromone present along edge } (i,j)$$

$$\eta(i,j) = \frac{1}{w(i,j)} : \text{“visibility”}$$

$$\alpha, \beta : \text{Parameters}$$

$$p(i,j) = \frac{(\tau(i,j))^{\alpha} \cdot (\eta(i,j))^{\beta}}{\sum_{k \in allowed} (\tau(i,k))^{\alpha} \cdot (\eta(i,k))^{\beta}}$$

Updating Pheromone Map

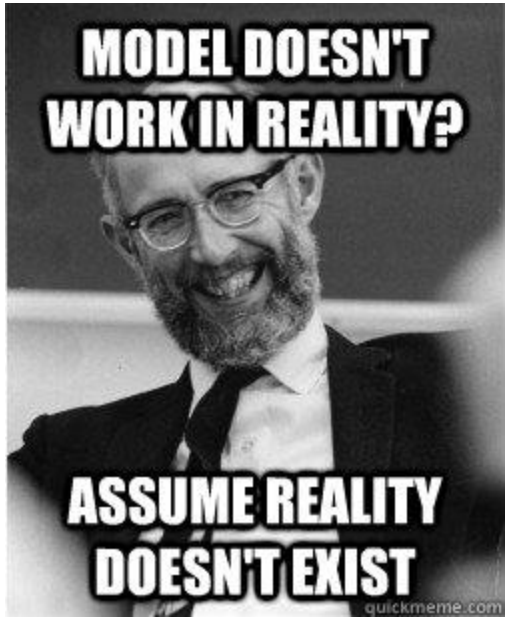
$$\rho : \text{Evaporation Constant}$$
$$\Delta\tau_{i,j} : \text{Density of pheromone laid along edge } (i,j)$$
$$Q : \text{Total amount of pheromone an ant can deposit}$$
$$L_k : \text{Length of } k\text{-th ant's tour}$$

$$\tau(i,j) \leftarrow \rho \cdot \tau(i,j) + \Delta\tau_{i,j}$$

$$\Delta\tau_{i,j} = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}$$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{Q}{L_k} \;, & \text{if } k\text{-th ant uses edge } (i,j) \\ 0 \;, & \text{otherwise} \end{cases}$$

How good can we model ants?



Tests on Western Sahara

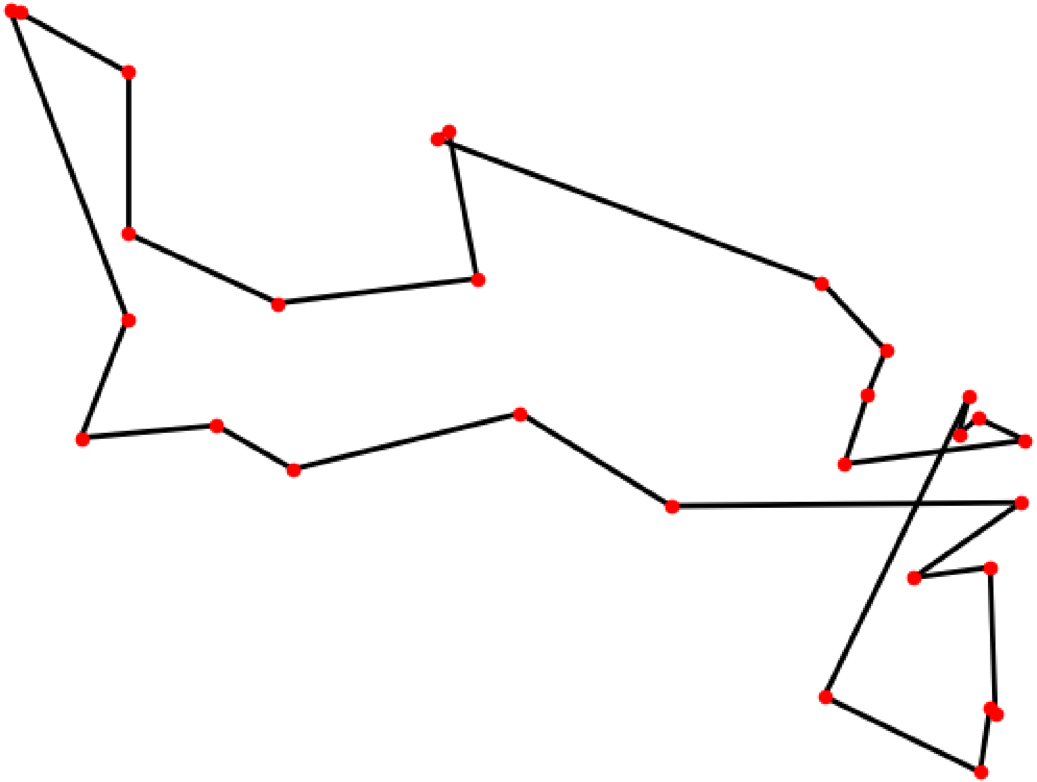
Vary the number of ants in the colony

$$\alpha = 1$$
$$\beta = 2$$
$$\text{number of iterations} = 500$$

Tests on Western Sahara

$$\text{ants} = 10$$

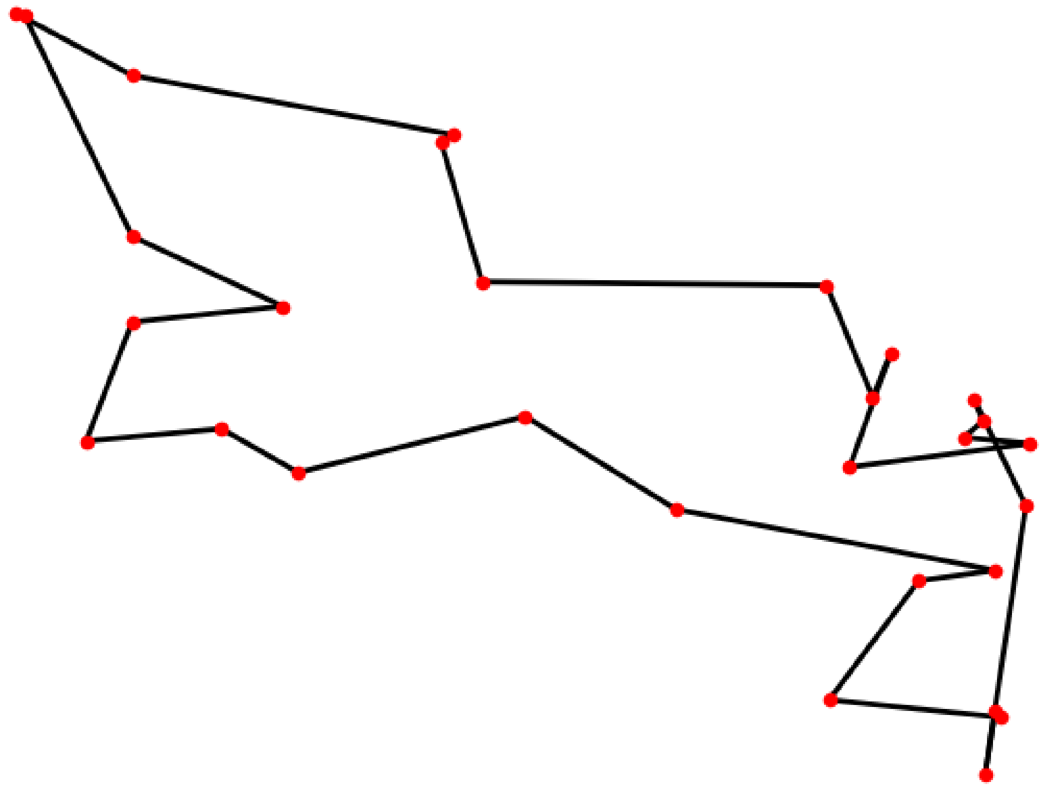
$$\text{distance} = 32,244$$



Tests on Western Sahara

$$\text{ants} = 20$$

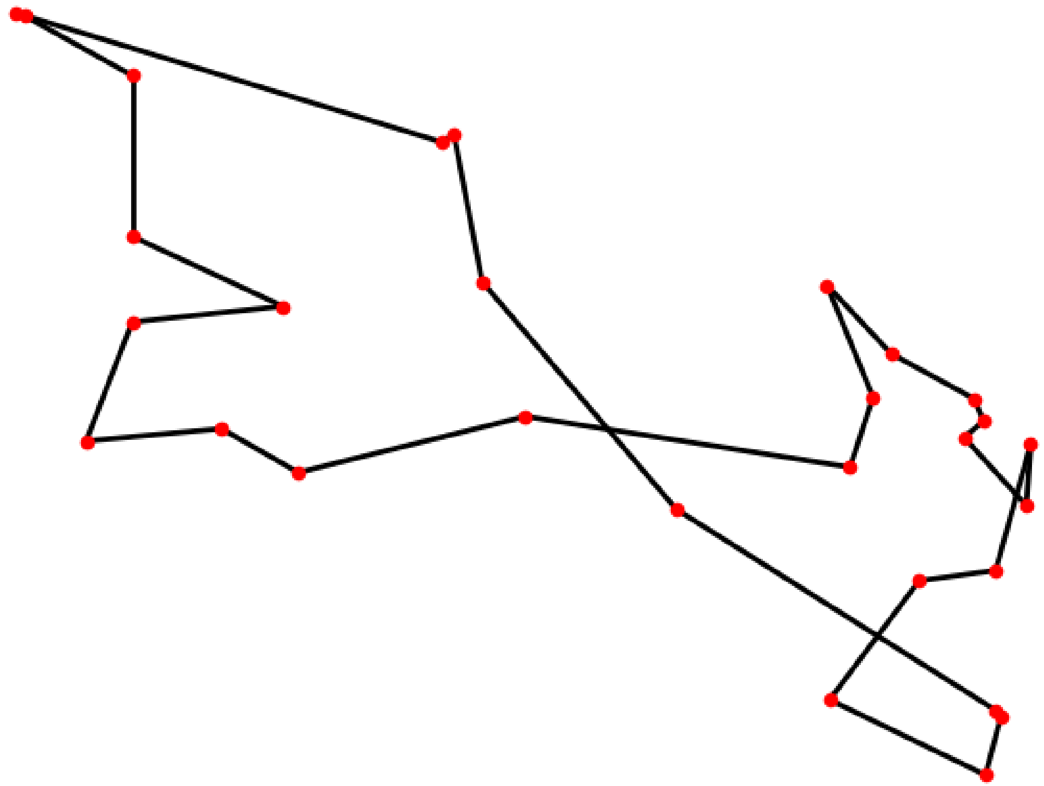
$$\text{distance} = 31,088$$



Tests on Western Sahara

$$\text{ants} = 30$$

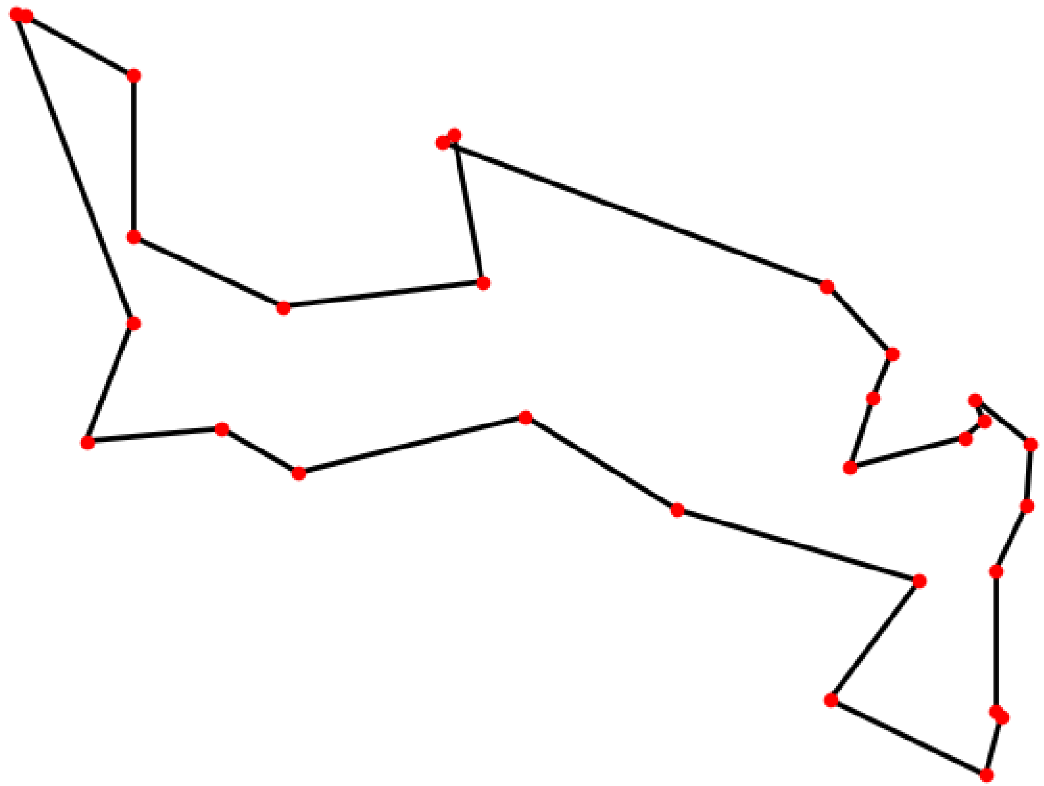
$$\text{distance} = 30,990$$



Tests on Western Sahara

$$\text{ants} = 50$$

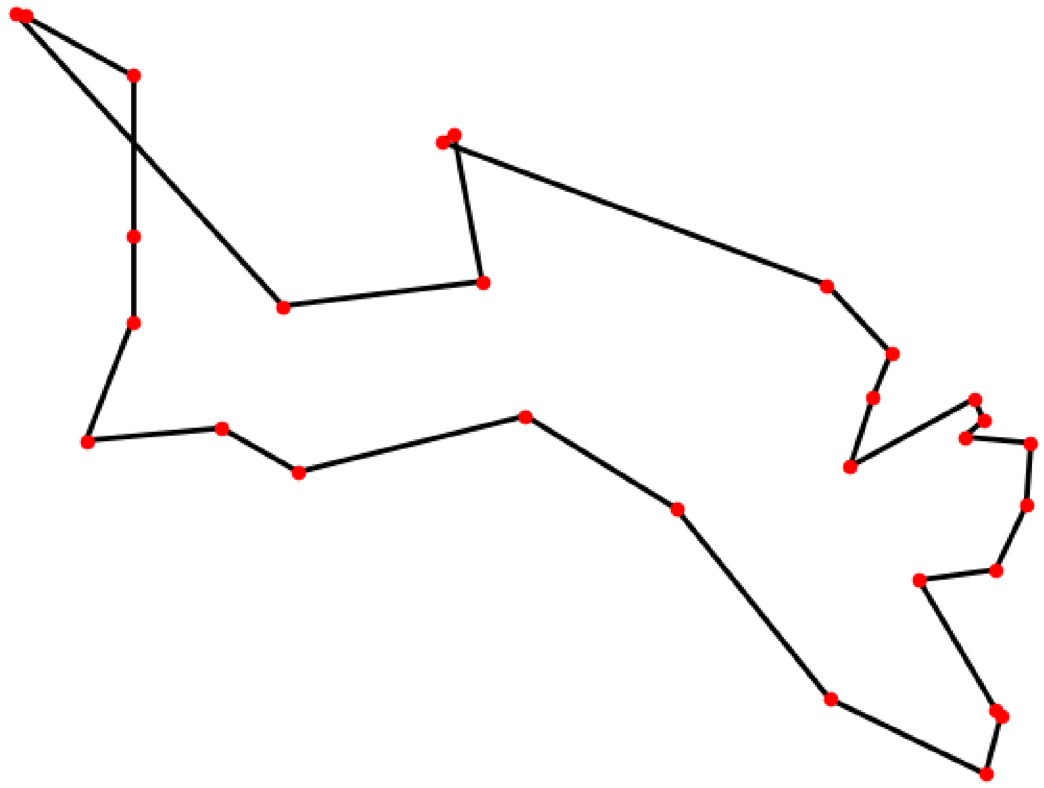
$$\text{distance} = 29,174$$



Tests on Western Sahara

$$\text{ants} = 100$$

$$\text{distance} = 28,762$$



Tests on Western Sahara

- Optimal tour is of length 27,603
- We found a tour of length 28,762

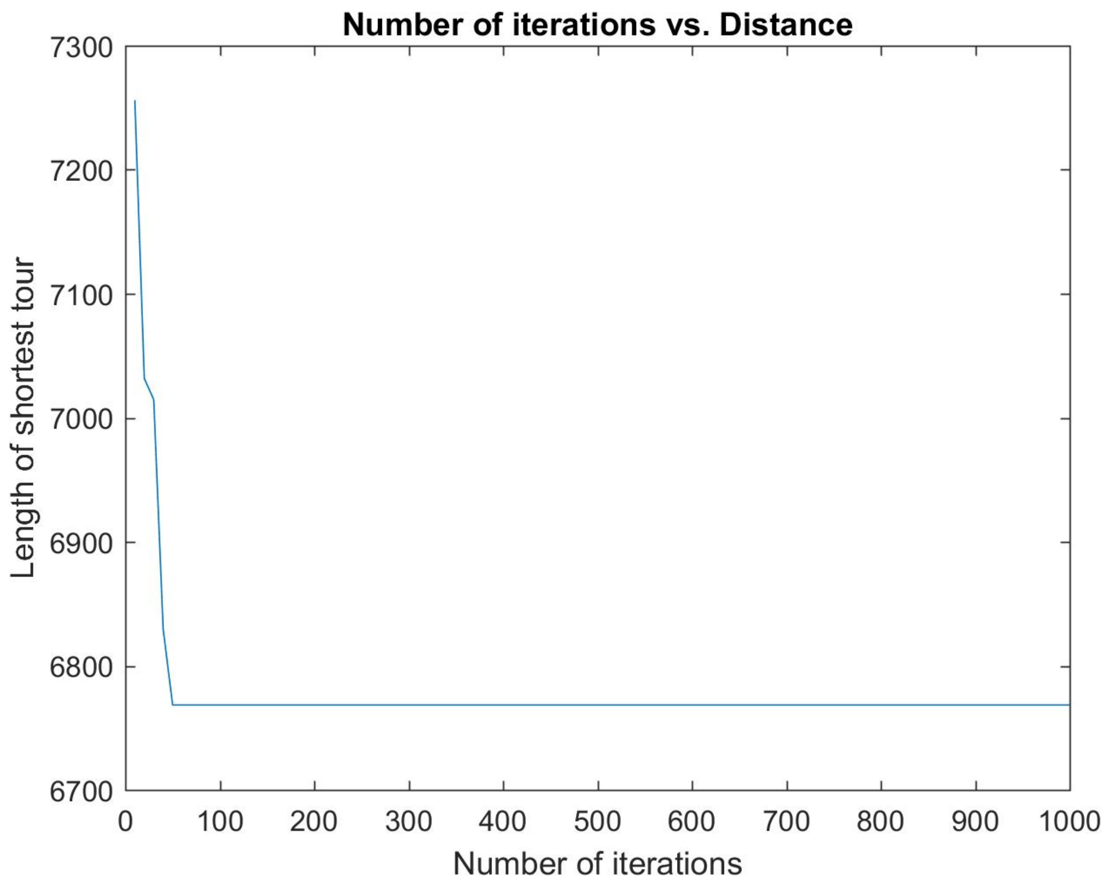
That's pretty good…

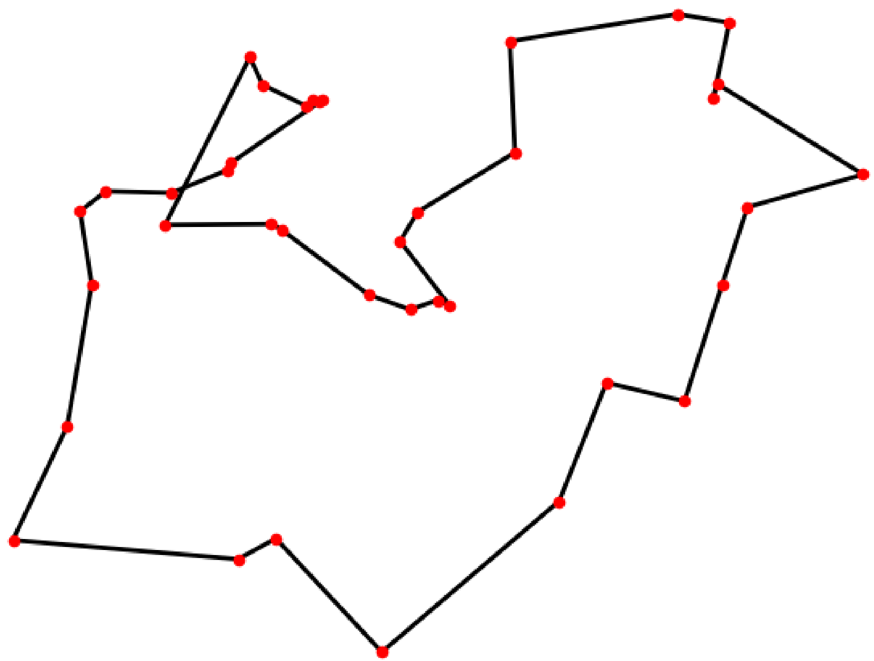
Tests on Djibouti

Vary the number of iterations

$$\alpha = 1$$
$$\beta = 2$$
$$\text{number of ants} = 30$$

Tests on Djibouti





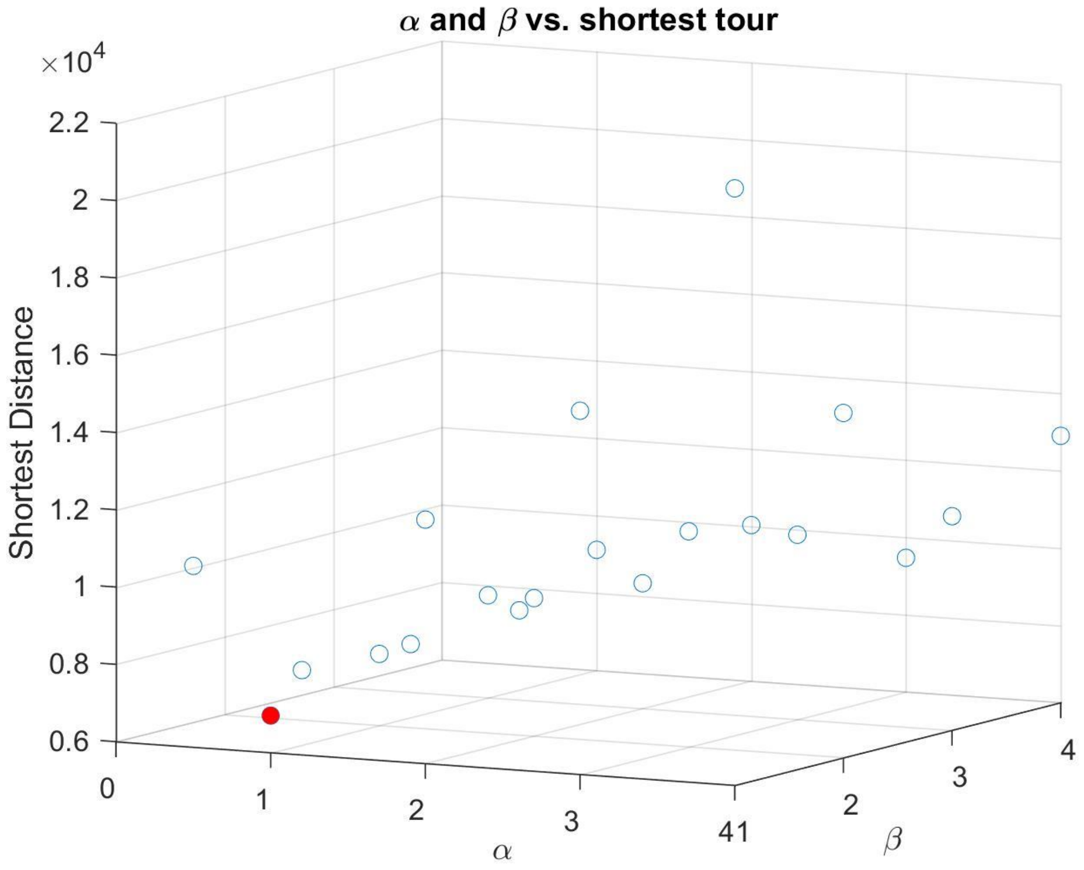
Tests on Djibouti

- Optimal tour is of length 6,656
- We found a tour of length 6,769

That is also pretty good…

But, can we do better?

Varying parameters (Djibouti)

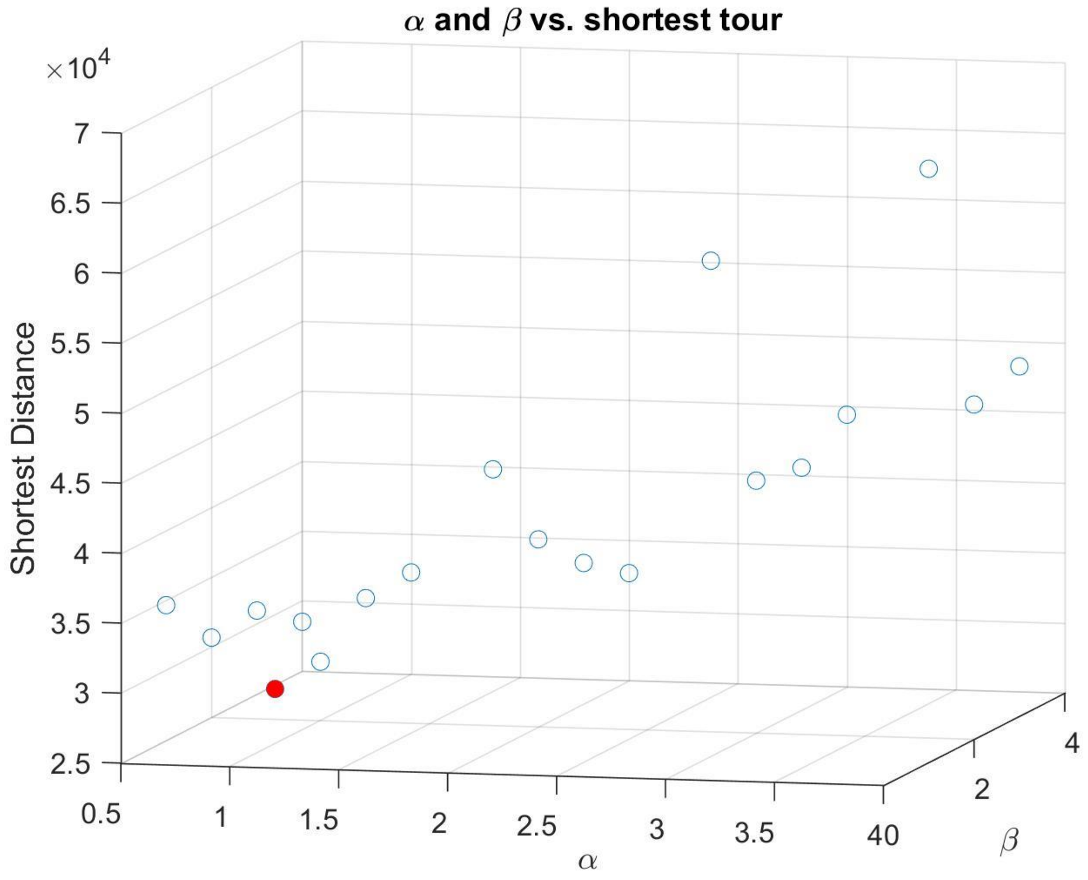


$$\alpha = 1$$

$$\beta = 1$$

$$\text{distance} = 6,955$$

Varying parameters (Western Sahara)



$$\alpha = 0.5$$

$$\beta = 4$$

$$\text{distance} = 28,503$$

What’s going wrong?

Drawbacks

- Too many parameters to tune
- No direct way to estimate the rate of convergence
- Due to its probabilistic nature, the algorithm may converge to the local optimum

The Future

Questions?