Travelling Salesperson Problem

Ádám Fischer
22203265
online presentation

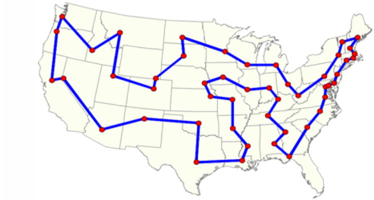
Data and Computational Science MSc

2023. 08. 04.

Introduction

- The problem statement starts with a simple question: given a list of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

- Alternative phrasing: what is the shortest Hamilton cycle of a graph?



P vs. NP

- The question can be viewed as a combinatorial optimization problem
- Why not list all possible routes and choose the shortest one?
  - TSP is a NP-hard problem
  - Since it is suspected (but never been proven) that NP $\neq$ P there is no algorithm that could solve a NP-hard problem in polynomial time
  - As the number of cities increases the problem becomes practically unsolvable just by checking each possible route

Applications

- Many practical applications mainly in logistics
- Appears in apparently remote areas such as DNA sequencing, manufacture of microchips

Proposed methods

1. Simulated Annealing
2. Ant Colony Optimization

Simulated Annealing

- Optimize a cost function $E(x)$ (distance traveled)
- Inspired by ideas from Physics: we need to view the optimization of $E(x)$ as the process of "cooling down" a physical system to zero temperature
- Simulated annealing uses a probabilistic approach to find the solution x_*, closely resembles the Metropolis–Hasting algorithm [1]

[1] Metropolis; Equation of state calculations by fast computing machines (1953)

Simulated Annealing

- We start with an initial (guess) state x_0 for the solution, and with an initial temperature T_0
- In every iteration we propose a new state for the solution, and if it's a better solution than the previous best guess, we accept it. If it's worse, we accept the propesed state with a given probability.
- As the systems "cools down" we will accept worse proposals for the solution with smaller and smaller probabilities
- This way the algorithm is able to escape from local optima and can find global optima

Pseudo code for Simulated Annealing

Algorithm 1 Simulated Annealing

Set initial state x_0 and $x = x_0$

Set initial temperature T_0

Set a cooling schedule T_k (somehow), that defines the number of iterations at each temperature

while stopping criterion is not met **do**

  propose new state x_k

  calculate $\Delta E = E(x_k) - E(x)$

  if $\Delta E < 0$ **then**

    accept the new state $x = x_k$

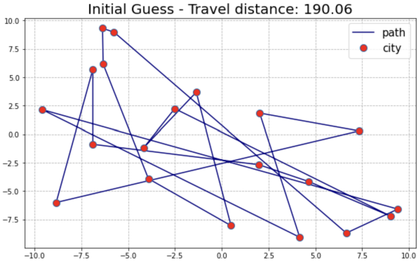
  else

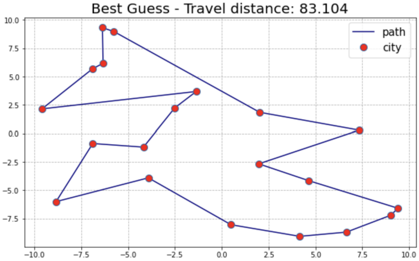
    accept the new state $x = x_k$ with probability $e^{-\Delta E / T_k}$

  end if

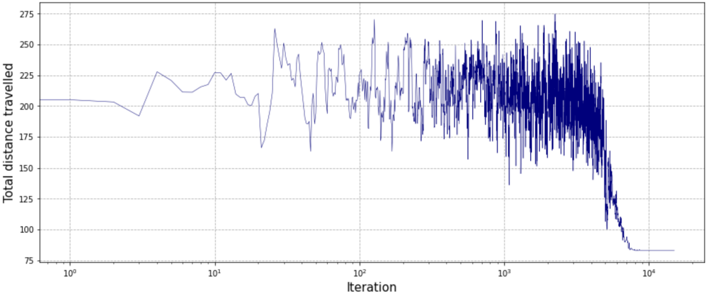
end while

Results I.


Initial Guess - Travel distance: 190.06
path
city
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

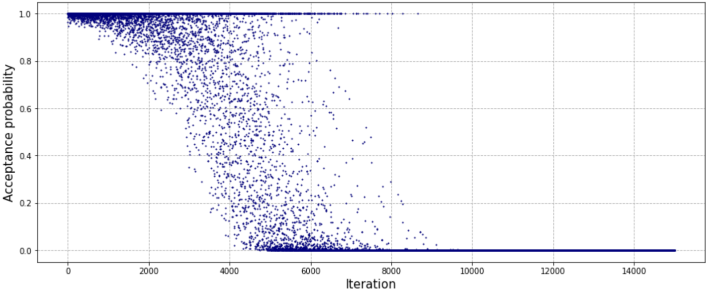

Best Guess - Travel distance: 83.104
path
city
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Results II.

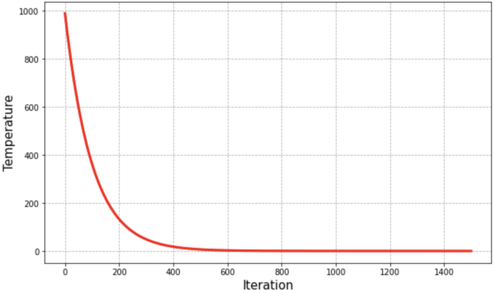


Total distance travelled
275
250
225
200
175
150
125
100
75
10^0
10^1
10^2
10^3
10^4
Iteration

Results III.



Acceptance probability
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
12000
14000
Iteration



Temperature
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
1400
Iteration

Ant Colony Optimization

- ACO method is inspired by the behavior of ants
- In real life, ants move randomly and leave pheromones as they go. Other ants are more likely to choose paths with more pheromones
- The pheromone left behind evaporates over time, so the pheromone density will be lower on paths that take more time for the ants to travel, compared to shorter paths $\Longrightarrow$ longer paths will be less likely to be chosen

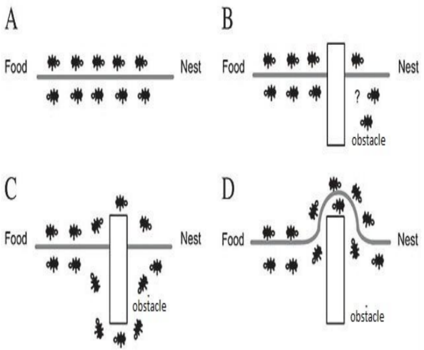


Figure: Pheromone trail is formed along the shorter path [2]

[2] Alok Bajpai, Raghav Yadav; Ant Colony Optimization For The Traveling Salesman Problem (2015)

Ant Colony Optimization

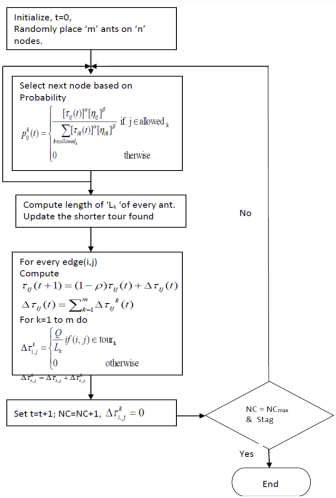
- Initially we place some ants on some nodes. Ant number k at iteration t will traverse from node i to node j with a probability of [2]

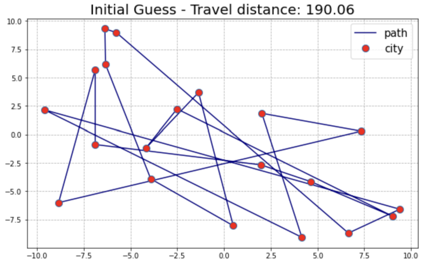
$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k \in allowed}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} & \text{if j} \in \text{allowed} \\ 0 & \text{otherwise} \end{cases}$$

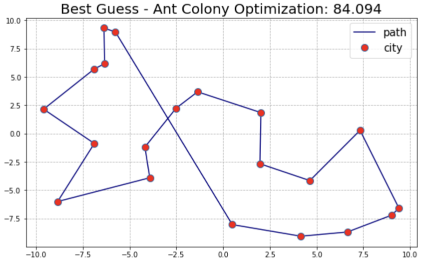
- where η_{ij} is the visibility of an edge (inversely proportional to the distence between node i and j)
- τ_{ij} is the pheromone concentration which is updated at the end of every iteration
- α and β are hyperparameters to be set

[3] Raghavendra; Solving Traveling Salesmen Problem using Ant Colony Optimization Algorithm (2015)

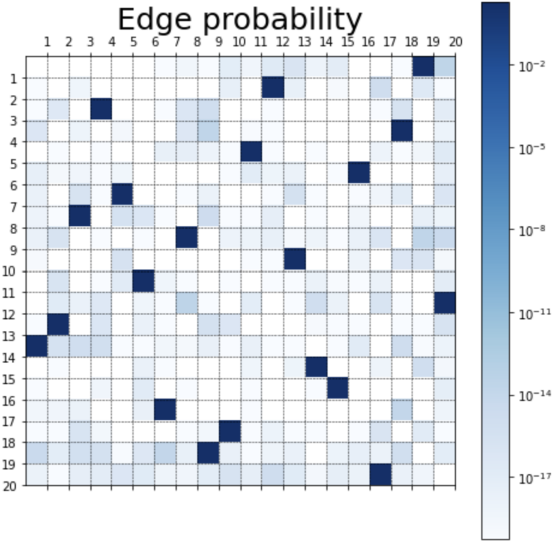
ACO flowchart




Initial Guess - Travel distance: 190.06
path
city
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Best Guess - Ant Colony Optimization: 84.094
path
city
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

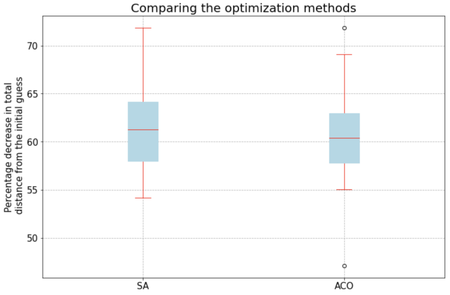
Results V.



Edge probability
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
10^{-2}
10^{-5}
10^{-8}
10^{-11}
10^{-14}
10^{-17}

Model comparison

- Hyperparameters set by hand (experince, recommendations in literature)



- Mean reduction with SA: 61.27%
- Standard deviation of reduction with SA: 4.88%
- Mean reduction with ACO: 60.5%
- Standard deviation of reduction with ACO: 5.9%

Thank you for your attention!