TU Clausthal

Meta-heuristic Optimisation Algorithms for Vehicle Routing

A Comparative Analysis

Despina Tawadros
Department of Informatics

23.03.2022

Agenda

- Introduction and Motivation
- Literature Review
- Research Scope and Limitations
- System Design
- Experimental Results and Analysis
- Summary
- Future Work

Introduction and Motivation

- The Vehicle Routing Problem (VRP) is one of the most studied combinatorial optimization problems and is concerned with the optimal design of routes.

Last-Mile Delivery
Final leg in the shipment process of the supply chain

Ride-sharing
A fleet of vehicles serving static and dynamic transportation requests

Freight-Integrated Transportation
Handling parcel requests along with passenger transportation requests

- Goal:

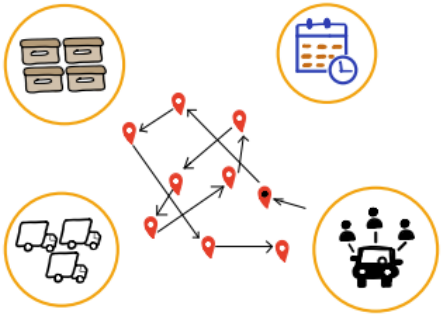


Fig. 1: VRP Objective & Constraints

- Challenge: NP-hard Problem
- Approach: Solving the VRP using Meta-heuristics

Literature Review

Swarm Intelligence for Solving a Traveling Salesman Problem

(Kuehner et. Al, 2020)

- Overview of 3 state-of-the-art Swarm Intelligence algorithms: Ant Colony Optimization, Particle Swarm Optimization & Bee Colony Optimization.
- Solving TSP
- Comparison of their performance
- ACO most accurate solution

Research on Optimization of Vehicle Routing Problem for Ride-sharing Taxi

(He Xu wt. Al, 2012)

- Proposed a routing optimization model for ride-sharing
- Static, capacitated vehicle routing problem with time windows
- Applying Simulated Annealing
- Tested on 9 network nodes and 29 clients
- 10 vehicles were sufficient 19% travel distance and 66% taxi demand spared

Genetic Algorithm Versus Ant Colony Optimization Algorithm

(Sariff et. Al, 2010)

- Comparison of performances in robot path planning applications
- Evaluation metrics: computational efficiency and accuracy
- Ant Colony Optimization converged faster and worked more accurately than the genetic algorithm

Research Scope

- Aims:
  - Efficiently create a route plan with
    - Accurate trip duration
    - Shortest trip duration
  - Compare the performance of three meta-heuristics in solving the VRP

- Objectives:
  1. Look for suitable datasets
  2. Perform feature engineering and data analysis
  3. Use a machine learning model to predict trip duration between any given pair of coordinates from the datasets
  4. Use the predicted durations to design a route plan using the three meta-heuristics
  5. Compare the performance of the three meta-heuristics based on the results obtained

Assumptions and Limitations

- Single vehicle assigned
- Static scheduling
- Subtours not allowed
- Every location is visited once
- No time or capacity constraints

→ Problem is reduced to TSP

Travelling Salesman Problem (TSP)

- Specific class of VRP
- Combinatorial Optimisation Problem
- NP-hard problem

- G = (V,E)

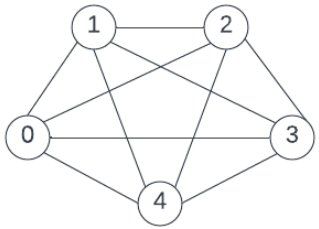


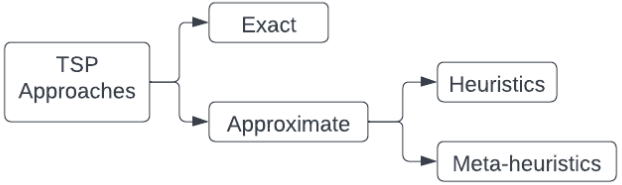
Fig. 2: TSP Graph

$$\begin{bmatrix} 0 & 1 & 3 & 4 & 5 \\ 1 & 0 & 1 & 4 & 8 \\ 3 & 1 & 0 & 5 & 1 \\ 4 & 4 & 5 & 0 & 2 \\ 5 & 8 & 1 & 2 & 0 \end{bmatrix}$$

Given
List of locations and distances between them

Goal
Find the order of cities to visit with the shortest path possible

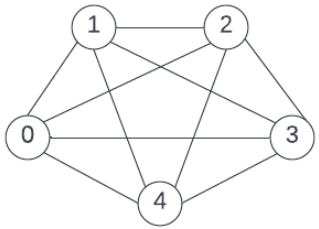
Conditions
- 1 Vehicle
- Visit each city once
- Subtours not allowed



Travelling Salesman Problem (TSP)

- Specific class of VRP
- Combinatorial Optimisation Problem
- NP-hard problem

- G = (V,E)

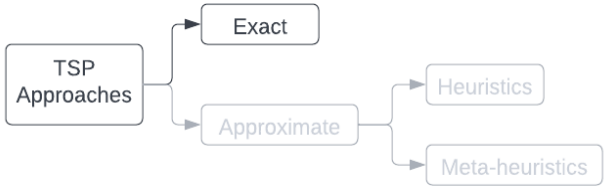


$$\begin{bmatrix} 0 & 1 & 3 & 4 & 5 \\ 1 & 0 & 1 & 4 & 8 \\ 3 & 1 & 0 & 5 & 1 \\ 4 & 4 & 5 & 0 & 2 \\ 5 & 8 & 1 & 2 & 0 \end{bmatrix}$$

Given
List of locations and distances between them

Goal
Find the order of cities to visit with the shortest path possible

Conditions
- 1 Vehicle
- Visit each city once
- Subtours not allowed



TSP - Exact Approaches

- Example: Brute-force algorithm
- NP-hard problem ⟶ cannot be solved in polynomial time
- As the number of nodes increases:
  - Permutations increase at a factorial rate
  - Computation time increases at an exponential rate O(n!)
- Scalability obstacle

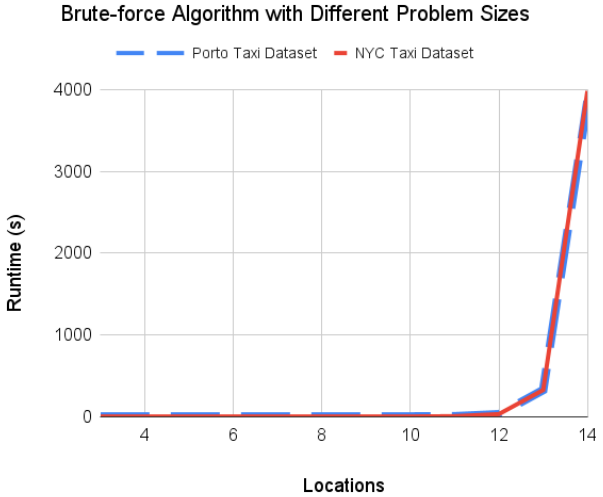
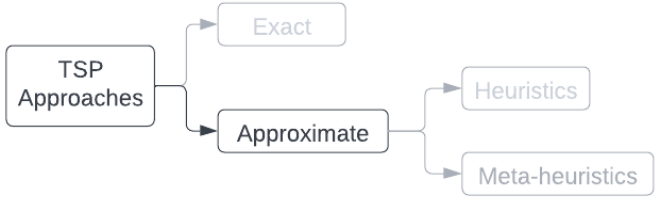
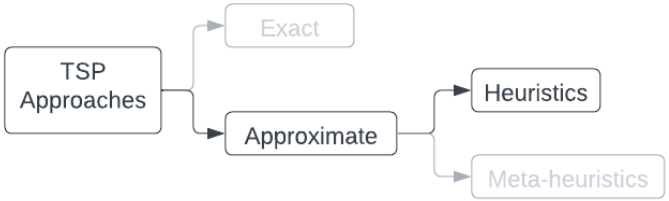


Fig. 3: Brute-force Algorithm



TSP – Approximate Approaches

- Aim:
  - Sufficiently good solution
  - Cut down computation costs
  - Reduce calculation time
- Tradeoff: ↓ solution quality ↑ calculation speed
- How? Explore a limited but promising sector of the solution space with high efficiency.
- Result: Near-optimal solution in polynomial time



Heuristics

Find the approximate solution then apply local improvements iteratively until the solution converges

Advantages

- Can find good solutions in short time when the search area is restricted
- Have no bounds on problem size and constraints as they can still find a near-optimal solution efficiently

Disadvantages

- Lack flexibility as they are usually problem-specific
- Get trapped more often in local minima because:
  - Greedy local optimal strategy by selecting the next node which would result in a local improvement but not global optimal solution
  - Intermediate solutions are not allowed to deteriorate during the search process of finding better solution

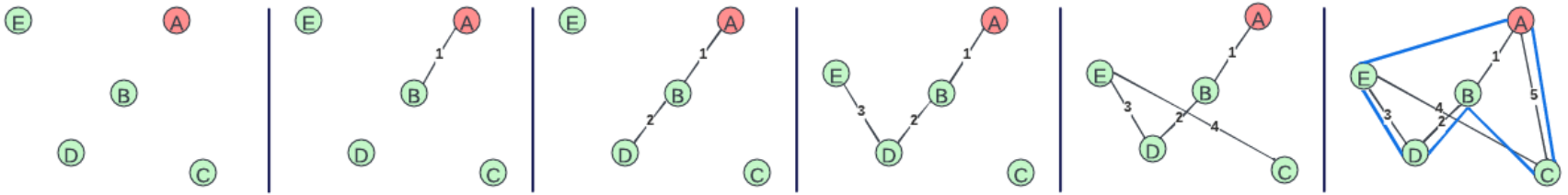
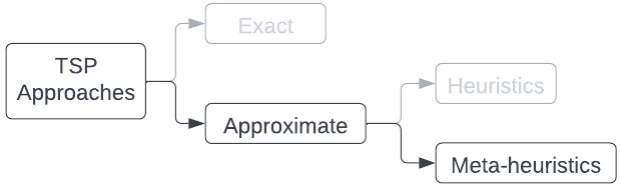


Fig. 4: Greedy local optimal strategy



Meta-heuristics

- Metaphor-based algorithms
- Strategies that guide an efficient process of solution space exploration
- Combine tour construction with tour improvement methods while using past enhancements with each iteration to build neighbourhood search processes for exploration and exploitation.
- Not problem-specific
- Non-deterministic
- Less often trapped in local minima by:
  - Temporarily accepting worse solutions
  - Employing local + population search methods
  - Performing a broader search on the solution space

Genetic Algorithm: A Brief Intro to Darwin‘s Survival of the Fittest

- Imitate the Darwinian analogy of survival of the fittest through natural selection to find near-optimal solutions
- Natural selection
  - Evaluate and select most fit candidate solutions
  - Apply small and slow gradual changes
- Survival of the fittest:
  - Least fit solution dies out, better ones continue to next iteration

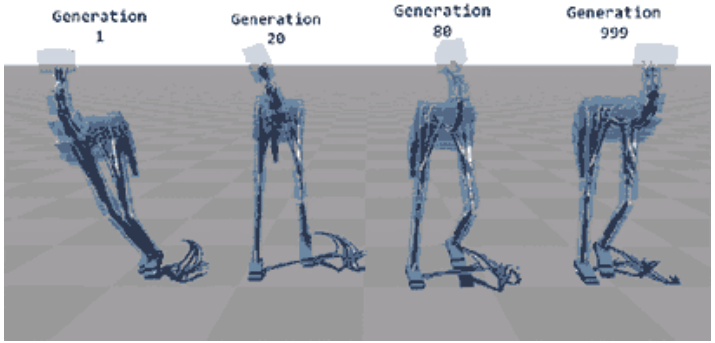
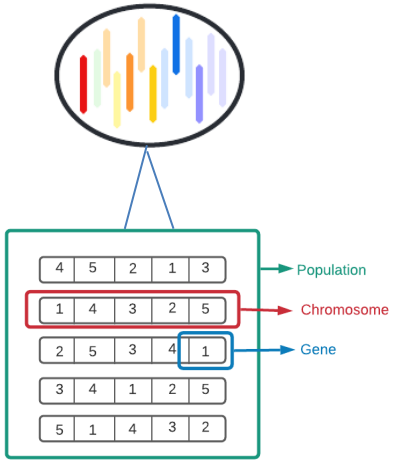


Fig. 5: GA-Survival of the fittest

Genetic Algorithm: Employing GA for TSP

1. Initialization

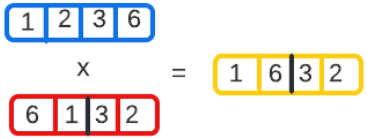


2. Evaluation & Selection

- Tournament Selection Technique
- Assignig a score to each individual
- Selection pressure

- Survival of Individuals with the highest fitness score
- Convergence rate controlled by selection pressure

3. Crossover



- A random point between a subtour on the first chromosome
- Passing down remaining alleles from second chromosome

- New generation with a mixture on non-identical yet superior genetic make-up

4. Mutation

- Purely random probabilistic change

- Local Convergence less likely

Ant Colony Optimisation Algorithm: Overview

- Swarm-based metaheuristic: collective behaviour of individuals influences the overall search behaviour
- Autonomous agents performing parallel local search processes
- Emulate the interaction or behaviour of ants finding the shortest path between their colonies and food source
- Pheromone levels deposited by ants guide other ants towards the shortest path to food source

Fig. 7: Ant in therapy

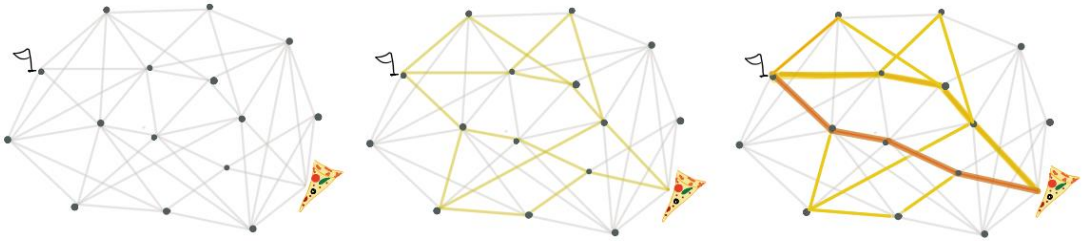


Fig. 6: ACO-Pheromone deposition

Ant Colony Optimisation Algorithm – Mathematical Formula

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{1}{L_k} & \text{k-th ant travels from i to j} \\ 0 & \text{otherwise} \end{cases}$$

- $\Delta\tau_{i,j}^{k}$: increase in pheromone amount
- L_k: total tour length traversed by the k^{th} ant

$$\tau_{i,j}^{k} = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} + (1-\rho)\tau_{i,j}$$

- $\tau_{i,j}^{k}$: total amount of pheromone deposited by all ants traversed i-j
- ρ: pheromone evaporation rate

$$P_{i,j} = \frac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum((\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta})}$$

- $P_{i,j}$: probability to pick edge i-j
- $\eta_{i,j}$: $1/d_{x,y}$ where d is the distance
- α, β: parameters that balance the influence of pheromone and desirability level

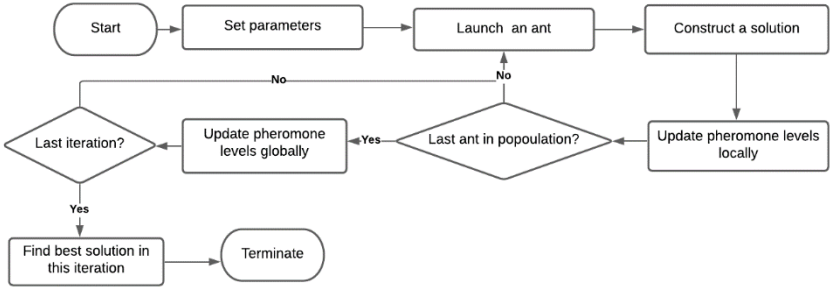
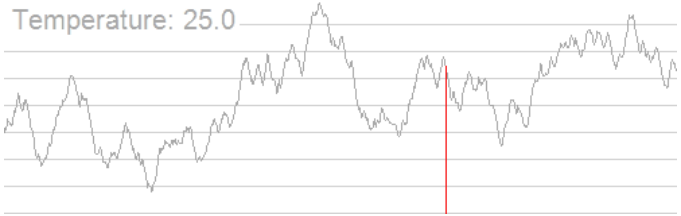


Fig. 8: ACO Flowchart

Simulated Annealing: Overview



Definition

- Class of Trajectory-based algorithms.
- Trajectory-based algorithms: search methods that operate on a single point in the solution space and iteratively improve the solution by exploring its neighbourhood for better solutions.
- Exploits the analogy of annealing.

How it Works

- Rapid heating of a material
  - ↳ Heating weakens molecular bonding
  - ↳ Material more ductile
- Gradual cooling down
  - ↳ Material gets harder to shape

How it relates to TSP

- The high temperature in the algorithms is equivalent to the measure of randomness by which changes are made to the path while exploring the solution space.
- Lowering the temperature gradually narrows the search range until the algorithm converges towards the global optimum.
- Therefore, high temperature increases the probability to accept worse solutions.
  - ↳ A wider space is explored
  - ↳ Decreases the risk of getting trapped in local optimum.

Simulated Annealing: Convergence

- Control Parameters:
  - Initial Temperature Tmax
  - End Temperature
  - Max. Number of Iterations
  - Maximum stay counter
- Acceptance probability: $r < e^{-\Delta c/t}$
- Candidate Generation Strategy: Swapping, Reversing or Transposing
- Cooling Schedule: $T(t) \sim 1/\log t$

The rate at which the maximum temperature decreases till it reaches the minimum temperature.

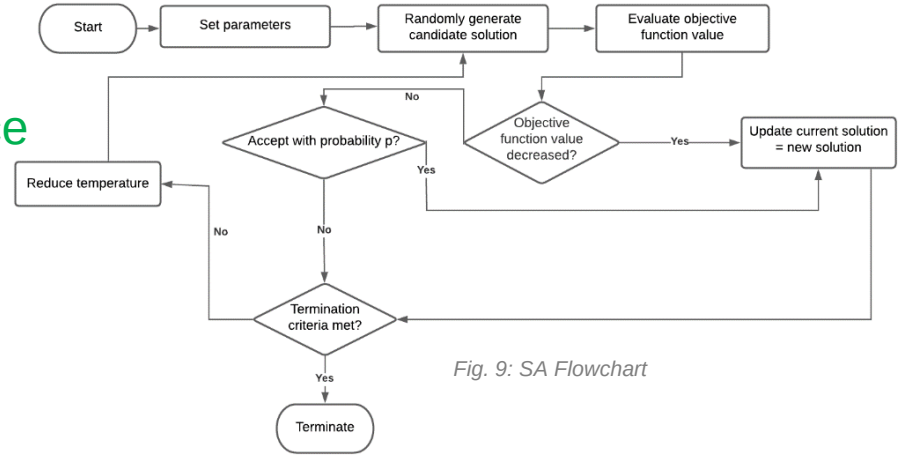


Fig. 9: SA Flowchart

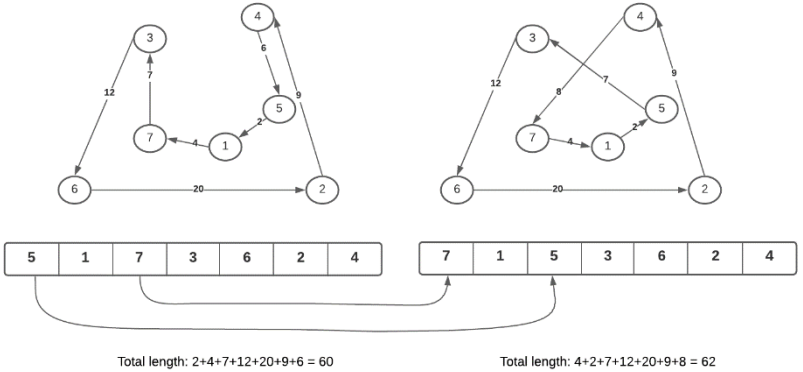
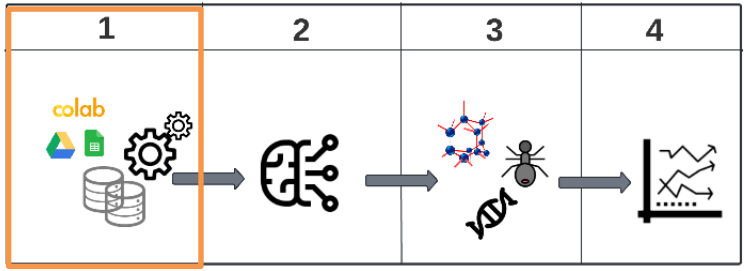
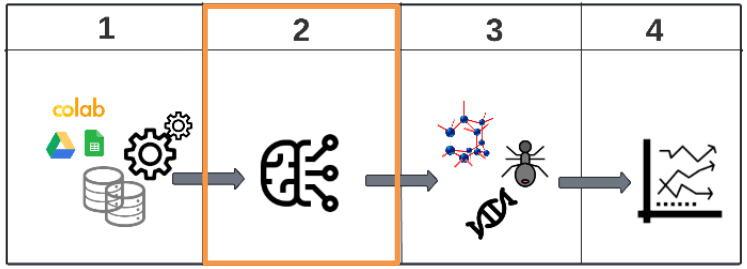


Fig. 10: SA Candidate generation strategy - Swapping

System Design –
1. Data Analysis and Pre-processing

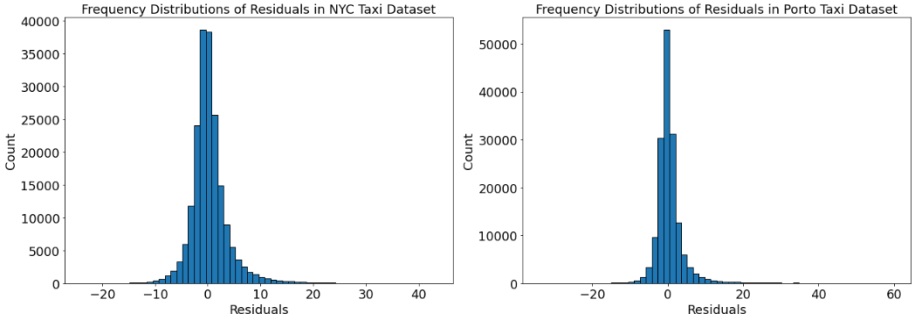


- Goal: Prepare and explore the necessary data for the experiments
- NYC Green Taxi Trip Records Dataset and Porto City Taxi Datasets
- Method:
  - Data filtering
  - Feature engineering
  - Statistical exploratory analysis
- Tools: Colab Notebook & Python 3
  - Colab provided 13 GB RAM and 2vCPU @ 2.2GHz.

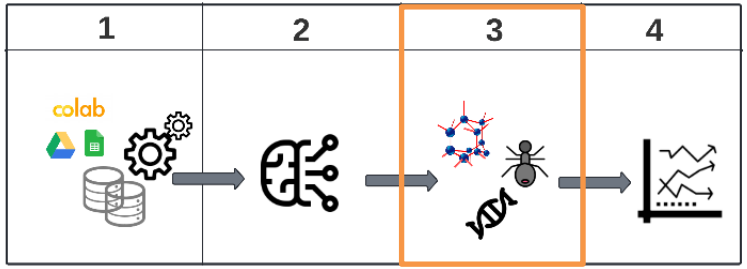


System Design –
2. Travel Time Forecasting

- Goal:
  - Predict the travel time between any given pair of locations from the datasets
- Method:
  - Trained a model using XGBoost on both preprocessed datasets using the following features: *pickup and dropoff coordinates, Manhattan distance, pickup weekday and pickup time*, where the target feature was *trip duration*.
- Model performance evaluation metrics: RMSLE, MAE and Residual Count graph
- Tools: XGBoost



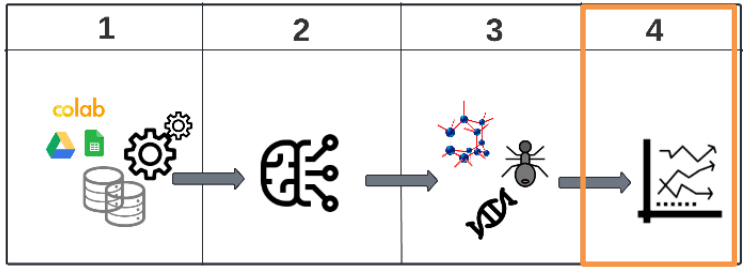
System Design –
3. Conducting Experiments using the three Meta-heuristics on both datasets



- Goal: Test the three Meta-heuristics using scikit-opt library
- Problem sizes : 25, 50, 75 and 100 locations
- Input: locations, cost matrix, algorithm-relevant parameter values
- Output:

```
Execution time: 1.2732350826263428 seconds
Final cost: [139.89976311] minutes, path: [2 0 4 1 3], for 800 generations
Final path addresses:
['West 145th Street & Saint Nicholas Avenue, West 145th Street, Manhattan '
 'Community Board 9, Manhattan, New York County, New York, 10031, United '
 'States',
 '15, Lawton Street, Brooklyn, New York, 11221, United States',
 '184, Bedford Avenue, Greenpoint, Brooklyn, New York, 11249, United States',
 'Church of the Good Neighbor, 115, East 106th Street, East Harlem, Manhattan '
 'Community Board 11, Manhattan, New York, 10029, United States',
 '137th Street Yard, West 141st Street, Manhattan Community Board 9, '
 'Manhattan, New York County, New York, 10031, United States']
```

System Design –
4. Experimental Analysis and Visualization (1)



- Goal: Compare the performance of the three algorithms on both datasets
- Method:

1. Defininig the optimal set of parameters for each problem size using grid search

Genetic Algorithm

- Population size
  - [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140]
- Number of generations
  - [600, 800, 1000, 1200, 1400, 1600, 1800, 2000]

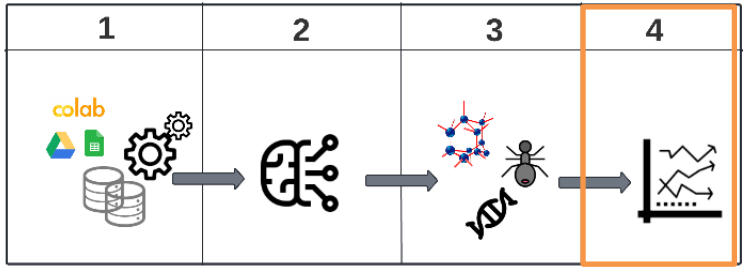
Ant Colony Optimization

- Population size
  - [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
- Number of iterations
  - [20, 300, 400, 500, 600, 800]
- Alpha & Beta
  - Permutations of [0.5, 1, 2, 3, 5]
- Evaporation coefficient = [0.1]

Simulated Annealing

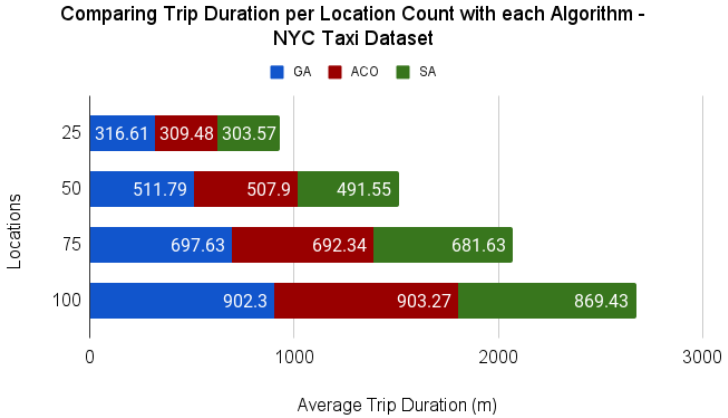
- Max. Temperature
  - [0.5, 1, 5, 10, 20, 30, 40, 50] °C
- Min. Temperature
  - [3^{-1}, 2^{-1}, 1^{-1}, 1^{-2}, 1^{-3}, 1^{-4}, 1^{-5}, 1^{-6}, 1^{-7}, 1^{-8}, 1^{-9}, 1^{-10}] °C
- Combinations of number of iterations at each temperature L : (n* locations) = [10, 20, 30, 40] with Maximum stay counter = [100, 200, 300, 400, 500]

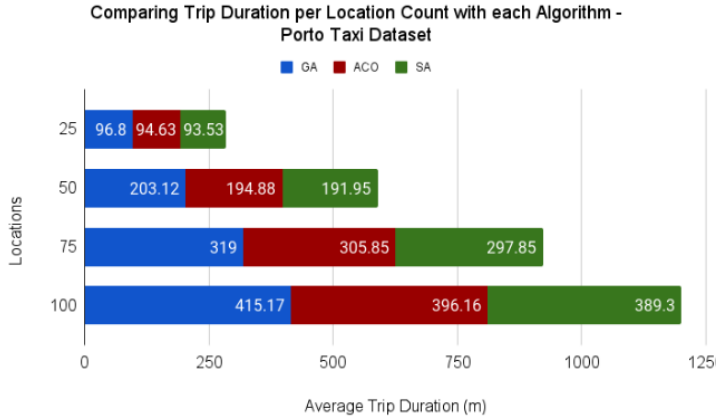
System Design –
4. Experimental Analysis and Visualization (2)



- Goal: Compare the performance of the three algorithms
- Method:
  1. Defininig the optimal set of parameters for each problem instance using grid search
  2. Repeating each experiment 5 times for each problem size to calculate the standard deviation and average
- Algorithms‘ evaluation metrics:
  - Average trip cost → Accuracy
  - Average runtime and convergence rates → Efficiency
  - Trip cost and runtime standard deviation → Reliability

Experimental Results & Analysis – Accuracy

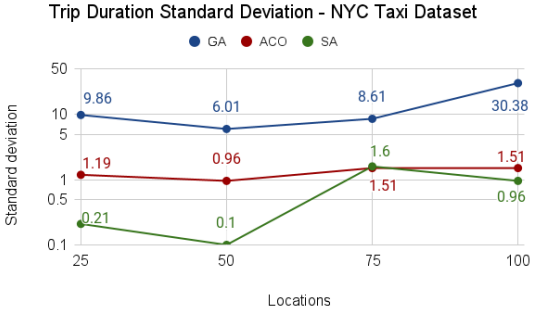


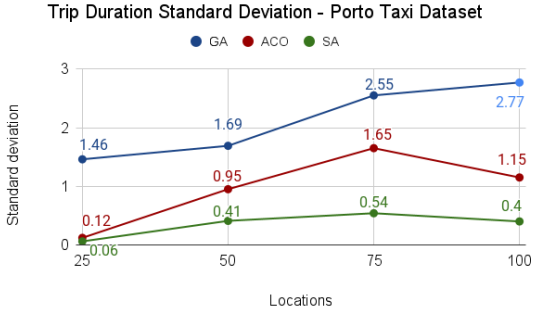


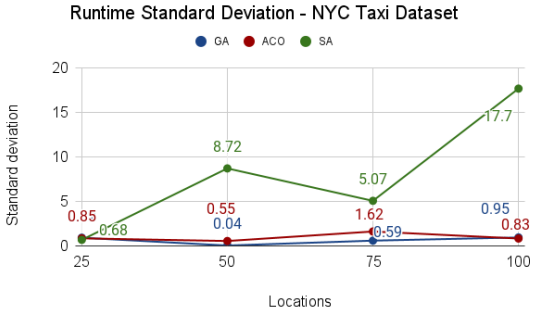
- SA has the shortest travel time
- Difference in calculated trip duratoin gets larger with larger problem sizes
- Solution accuracy drastically decreases with larger problem sizes with GA

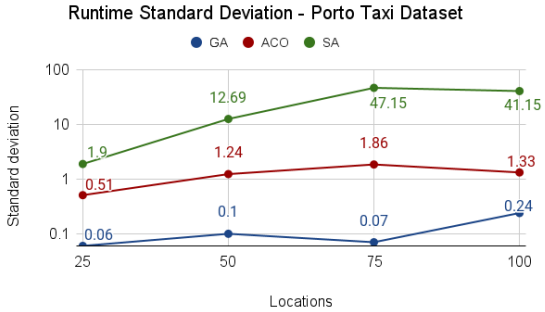
Experimental Results & Analysis – Reliability

- SA most reliable & GA least reliable result
- Insignificant variations in runtime with GA & ACO
- SA runtime extermely unstable with problem size starting from 50



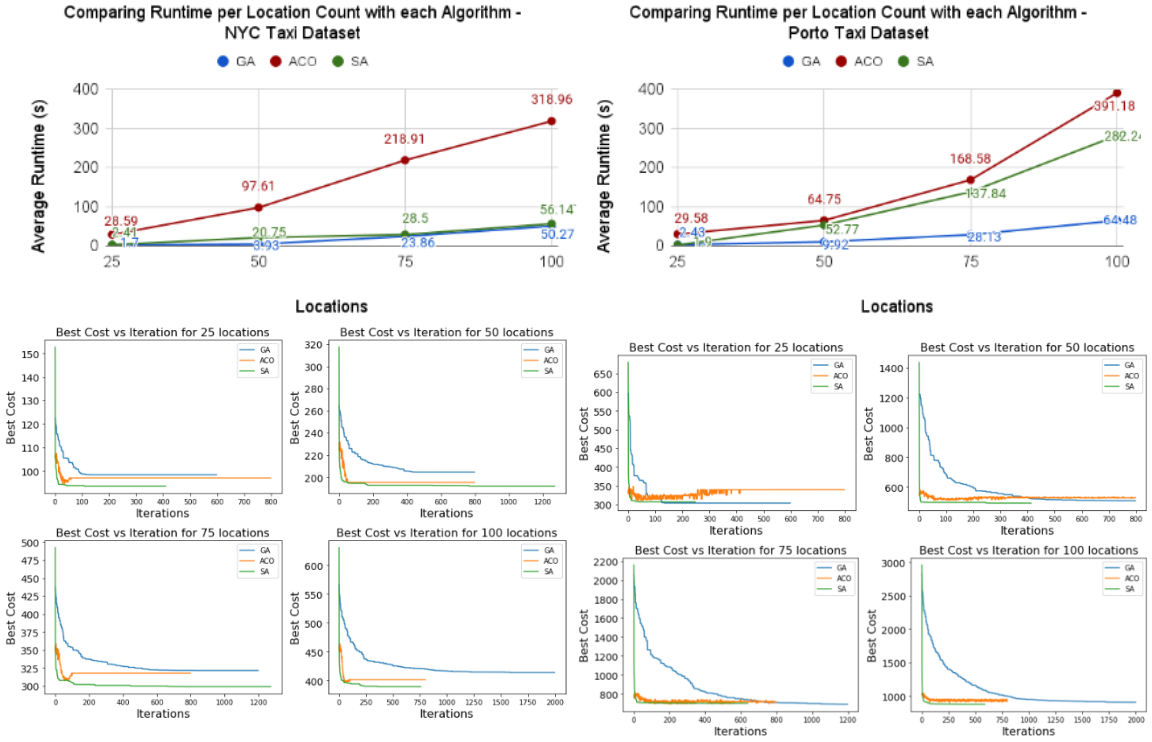






Experimental Results & Analysis – Efficiency

- ACO most inefficient in both datasets
- GA efficient over both datasets
- SA more stable with NYC Dataset
- SA shows fastest convergence rate
- ACO unstable convergence behaviour
- GA prematurely converges



Conclusions

	Solution Accuracy	Reliability (Result)	Reliability (Runtime)	Efficiency	Implementation
GA	3	3	1	1	1
ACO	2	2	2	3	3
SA	1	1	3	2	2

Summary

- Covered:
  - 2 Datasets were pre-processed and analysed
  - Employed a machine learning model to estimate the trip duration between any location pair
  - Applied 3 different meta-heuristc algorithms in solving the TSP
  - Pipeline developed and expermental setup
- Tested:
  - Performance of GA, ACO and SA on two datasets with four different problem sizes
- Concluded:
  - Solution accuracy and reliability > Computational Resources → SA
  - Resources and Runtime > Optimal reliable Solution → GA

Future Work

- Expand the simple TSP scenario by applying more constraints such as:
  - Time window
  - Vehicle Capacity
  - Max. detour time

- Test hybridising optimization approaches:
  - Combining two meta-heuristics
  - Combining one heuristic with one meta-heuristic

Generalizations

- Design of global navigation satellite system (GNSS) surveying networks (Saleh & Chelouah, 2004)
  - Space-based satellite system providing coverage for all locations worldwide
  - Early warning and management for disasters
  - Determine the geographical positions of unknown points on and above the earth using satellite equipment
  - When there are multiple receivers or multiple working periods, the problem of finding the best order of sessions for the receivers can be formulated as an mTSP
- Computer wiring (Lenstra & Rinnooy Kan, 1974)
  - Connecting components on a computer board
  - Modules are located on a computer board and a given subset of pins has to be connected
  - Finding a shortest Hamiltonian path with unspecified starting and terminating points
- Imaging celestial objects - NASA *Starlight* Space Interferometer Program
  - Minimize the use of fuel in targeting and imaging maneuvers for the satellites involved in the mission (the cities in the TSP are the celestial objects to be imaged, and the cost of travel is the amount of fuel needed to reposition the satellites from one image to the next)

References

- A. Adewole, K. Otubamowo, T. Egunjobi, and K. Ng. “A Comparativ Study of Simulated Annealing and Genetic Algorithm for Solving the Travelling Salesman Problem”. In: International Journal of Applied Information Systems 4 (Oct. 2012), pp. 6–12.
- L. Altenberg. “Evolutionary Computation”. In: Encyclopedia of Evolutionary Biology. Ed. by R. M. Kliman. Oxford: Academic Press, 2016, pp. 40–47. isbn: 978-0-12-800426-5.
- S. Angus, M. Holger, F. W, K Phang, H Seah, and C Tan. “Selection of Parameters for Ant Colony Optimisation Applied to the Optimal Design of Water Distribution Systems”. In: Dec. 2001
- A. M. S. Asih, B. M. Sopha, and G. Kriptaniadewa. “Comparison study of metaheuristics: Empirical application of delivery problems” In: International Journal of Engineering Business Management 9 (2017), p. 1847979017743603.
- Bayram, Hüsamettin, ¸Sahin, and Ramazan. “A new simulated annealing approach for the traveling salesman problem”. In: Mathematical and Computational Applications 18 (Dec. 2013)
- Matai, R., Singh, S., Mittal, M. L. , 2010, 'Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches', in D. Davendra (ed.), Traveling Salesman Problem, Theory and Applications, IntechOpen, London. 10.5772/12909.
- Z. Chen and W. Fan. “A Freeway Travel Time Prediction Method Based on an XGBoost Model”. In: Sustainability 13 (July 2021) S. Desale, A. Rasool, S. Andhale, and P. Rane. “Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey”. In: INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS 351 (Jan. 2015)
- M. Dorigo, M. Birattari, and T. Stutzle. “Ant colony optimization”. In: IEEE Computational Intelligence Magazine 1.4 (2006), pp. 28–39
- M. Dorigo and L. M. Gambardella. “Ant colonies for the travelling salesman problem”. In: Biosystems 43.2 (1997)
- A. Gharib, J. Benhra, and M. Chaouqi. “A performance comparison of PSO and GA applied to TSP”. In: International Journal of Computer Applications (Oct. 2015)
- http://www.math.uwaterloo.ca/tsp/apps/index.html (University of Waterloo, January 2007)
- Devin Soni. Introduction to Evolutionary Algorithms - Optimization by natural selection 2018. url: https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac

Thanks for your attention !

Questions?

Motivation

Description	Issues	Resolution	Goal
• Final step in the shipment process of the supply chain • Delivering parcels from transportation hub to final destinations • Trip routes planned in advance	• Most expensive and inefficient part of the shipping process • 53% of the whole shipment costs and 41% of the overall supply chain costs • Less sustainable	• Profiting from urban transportation by incorporating parcel delivery into ride-sharing • Integrating pick up and delivery of parcels with passenger transportation requests • Inefficient route planning remains an unsolved issue	• Cut delivery times and transportation costs • Meta-heuristics in route planning • Integrating pick up and delivery of parcels with passenger transportation requests

Experimental Setup

Problem Sizes = {25, 50, 75, 100}
The following hyperparameter values were tuned with each problem size:

Genetic Algorithm

- Population size
  - [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140]
- Number of generations
  - [600, 800, 1000, 1200, 1400, 1600, 1800, 2000]

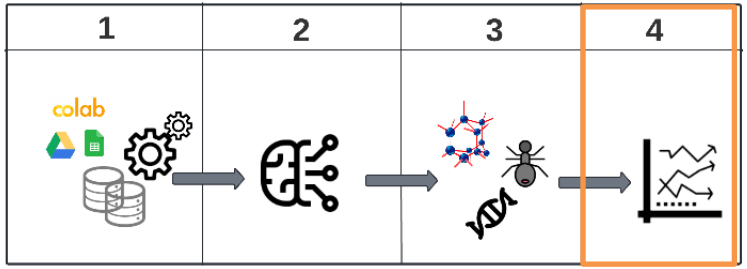
Ant Colony Optimization

- Population size
  - [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
- Number of iterations
  - [20, 300, 400, 500, 600, 800]
- Alpha & Beta
  - Permutations of [0.5, 1, 2, 3, 5]
- Evaporation coefficient = [0.1]

Simulated Annealing

- Max. Temperature
  - [0.5, 1, 5, 10, 20, 30, 40, 50] °C
- Min. Temperature
  - [3^{-1}, 2^{-1}, 1^{-1}, 1^{-2}, 1^{-3}, 1^{-4}, 1^{-5}, 1^{-6}, 1^{-7}, 1^{-8}, 1^{-9}, 1^{-10}] °C
- Combinations of number of iterations at each temperature L : (n* locations) = [10, 20, 30, 40] with Maximum stay counter = [100, 200, 300, 400, 500]

System Design –
4. Experimental Analysis and Visualization



- Goal: Compare the performance of the three algorithms
- Method:
  - Defininig the optimal set of parameters for each problem instance using grid search
  - Repeating the experiments 5 times for each problem size to calculate the standard deviation and average
- Evaluation metrics:
  - Average trip cost → Algorithm Accuracy
  - Average runtime and convergence rates → Algorithm efficiency
  - Trip cost and runtime standard deviation → Algorithm reliability

Conclusions

GA

- Simplest to implement
- Single parameter to adjust
- Most efficient algorithm
- Stable
- Poorest performance in terms of solution accuracy

SA

- Most accurate and most reliable results
- Inconsistent computation time with large problem instances
- Iteration parameters ensure convergence

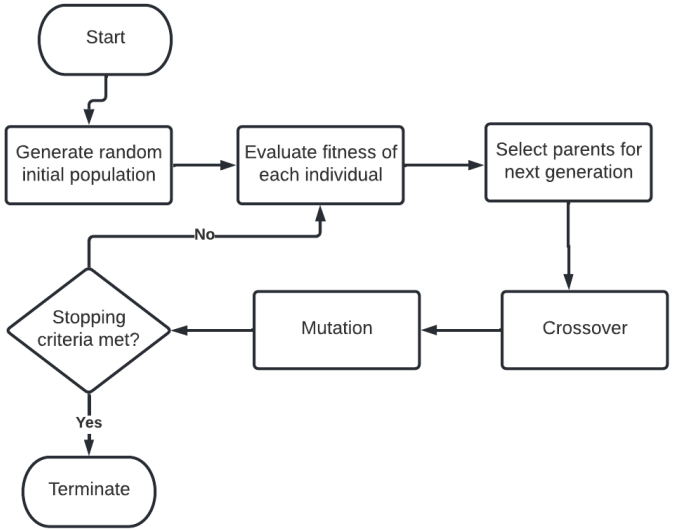
ACO

- High sensitivity to parametric variations
- Tuning three interdependent parameters
- Relatively good results in small to medium-sized instances
- Pheromone evaporation mechanism counteracts the impact of the randomized initial solution
- Unstable convergence

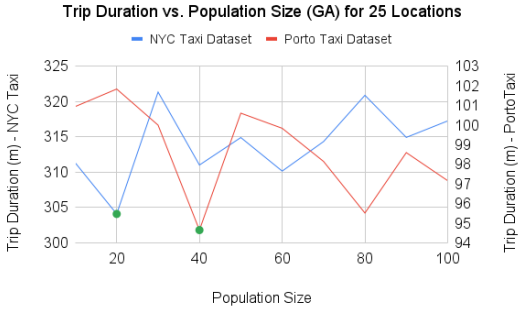
Backup Slide – Behaviour Explanation

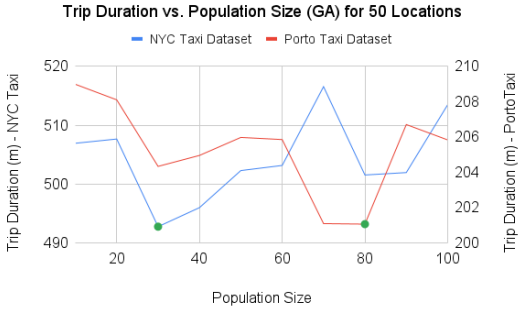
- For GA, there is no rules has been used to initialize the population where it based on random approaches. Effect from this process, the population in the initial generations will consists of optimal and non optimal path that will generate variety values of distance. This will cause the range of distance to be optimized by GA is bigger than ACO. Moreover, although GA will carry forward the optimal path to the next generation during the selection process, the possibility to obtain the population consists of non optimal child will repeated again. This is because the point to cross and mutate the chromosomes also will determine randomly and thus cause to the increment of the optimized data. As a result, it shows that the random initialization process of GA from one generation to other generation had cause GA to optimize the wide range of distance.
- ACO initializes the population by using a state transition rules is more efficient compared to GA that is based on random approaches.

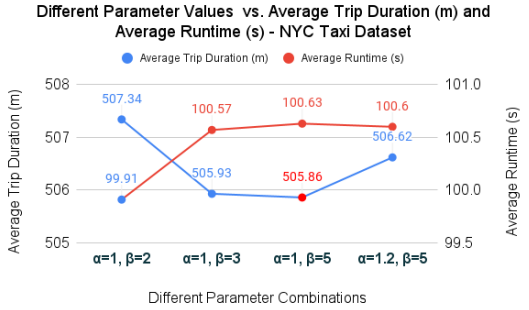
Backup Slide – GA Flowchart

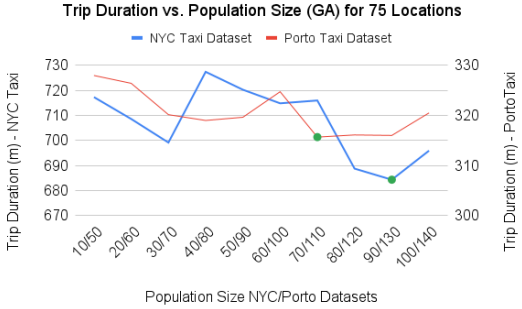


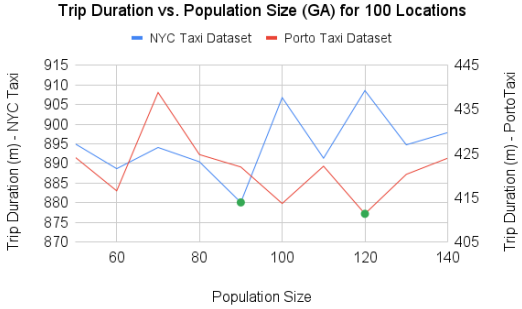
Backup Slide – GA and ACO Parameter tuning example

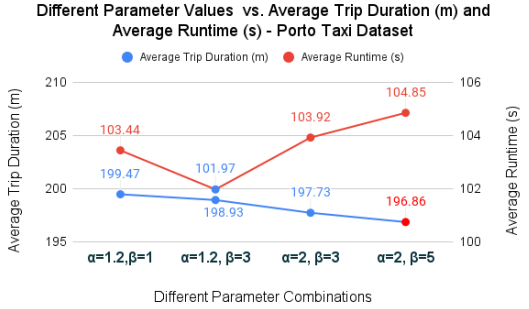




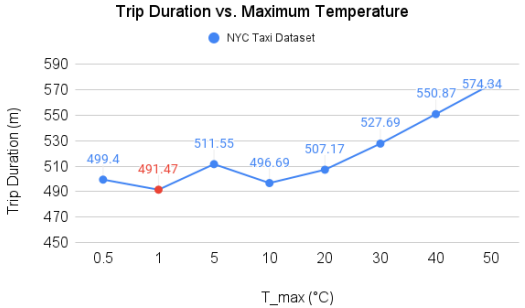


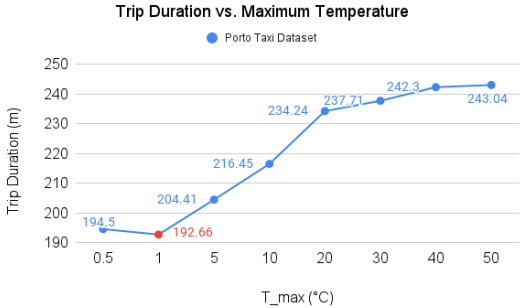


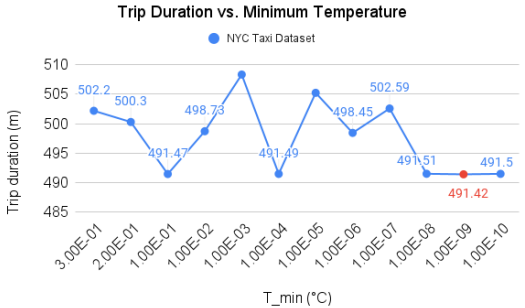


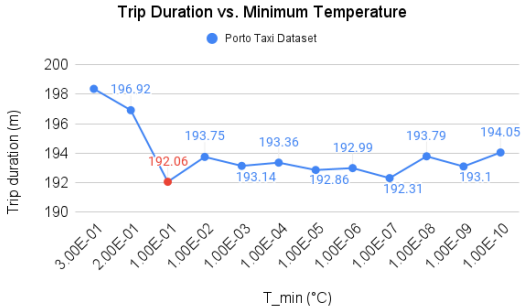


Backup Slide – SA Parameter tuning example









Backup Slide – SA Parameter tuning example

	NYC dataset	Porto dataset
RMSLE	0.0514	0.0633
MAE	2.340	2.203
Max. Error	43	60