Travelling Salesman problem Using genetic Algorithm

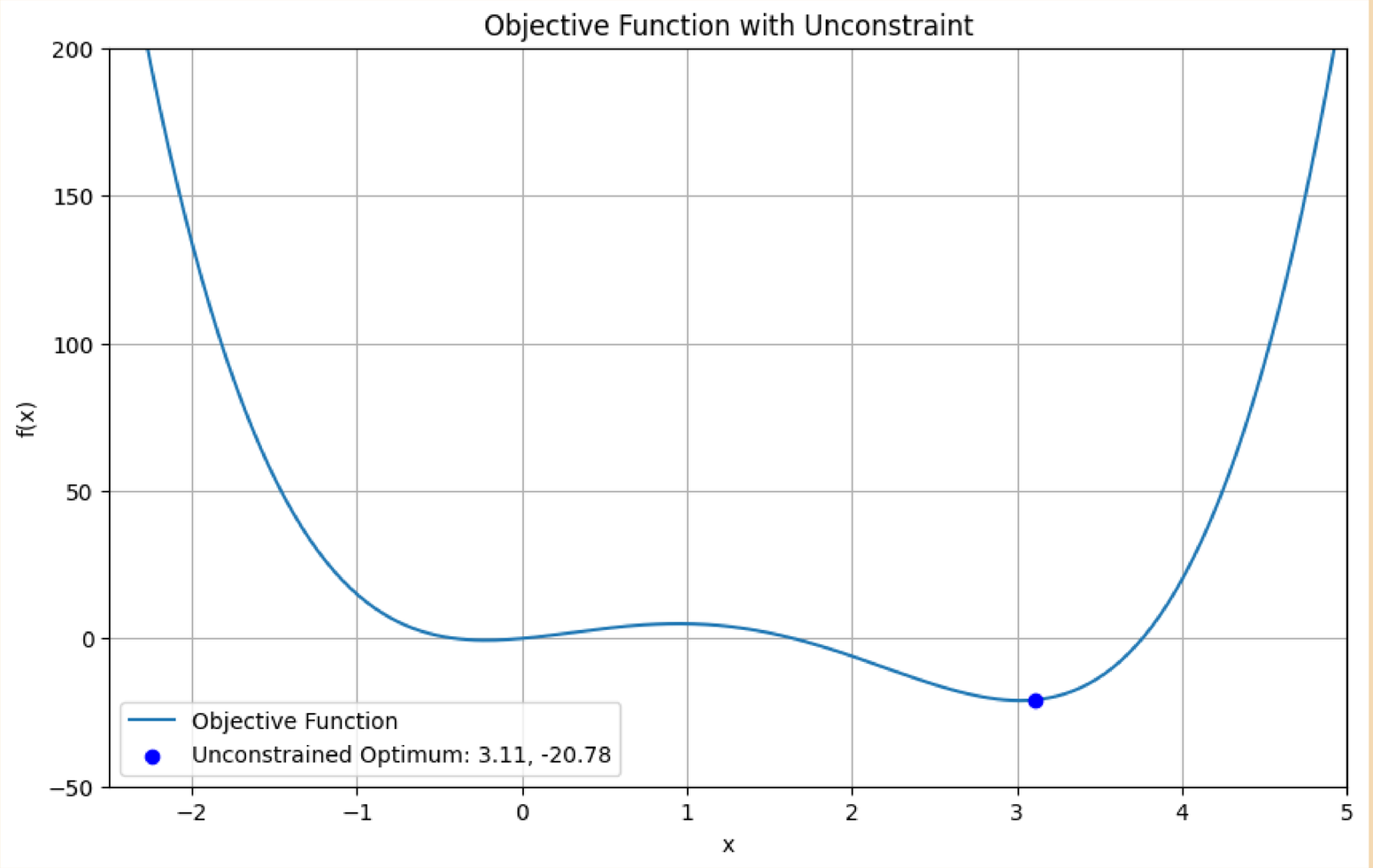
Algorithm Spring2024

Presented by: 박상범(21800272), 방은주(22100331), 황은지(22100809), 구나영(22101006)

PRESENTATION OUTLINE

- Understanding Genetic Algorithm (Task 1)
- Find the Shortest Route for the United Parcel Service (UPS)
- Find the Best Route for BLACKPINK World Tour

UCONSTRAINT

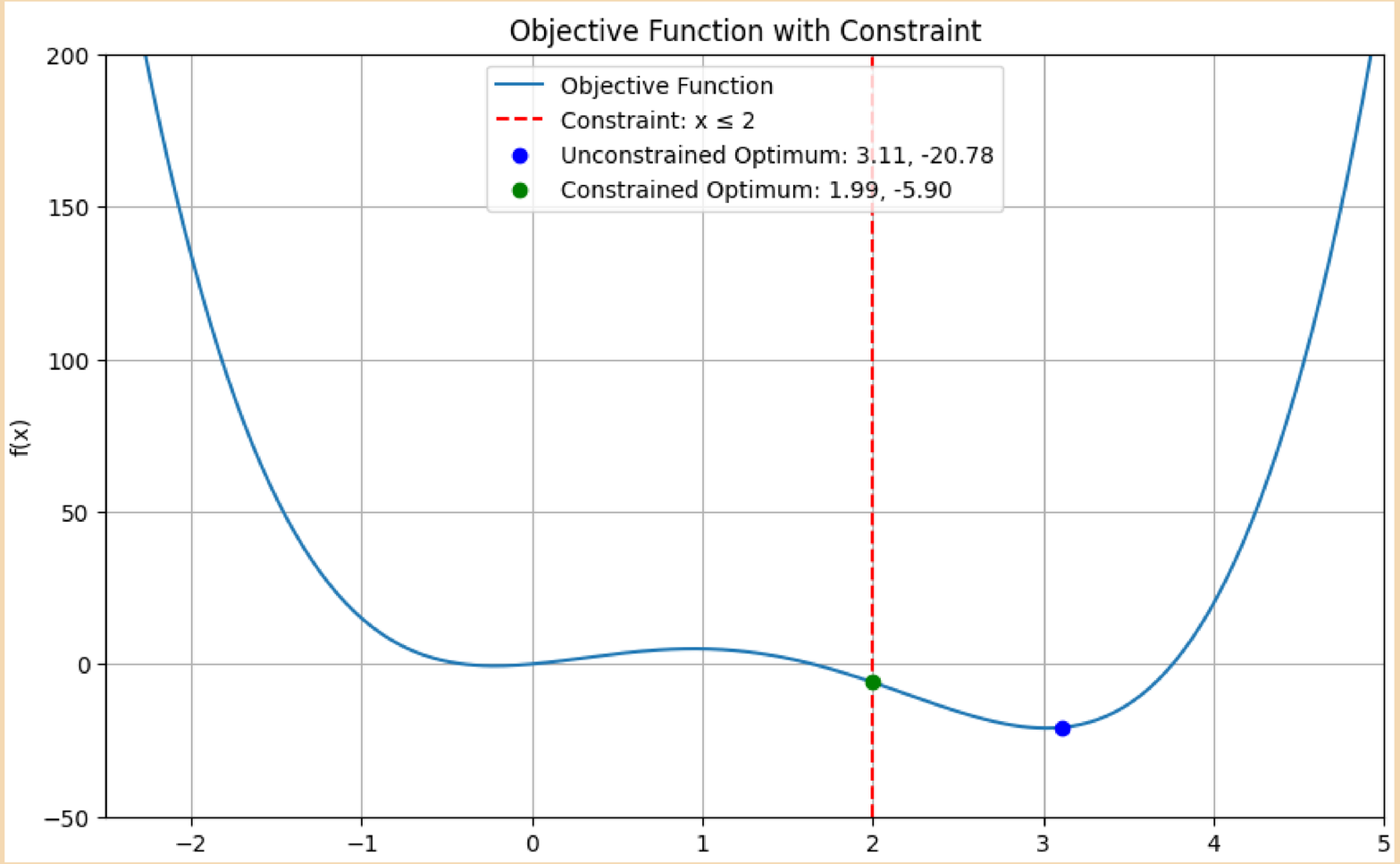


- Uncontraind data: 3.0323751244638713 , -21.00464598622979

Q. How did you design the stopping criteria and what is the optimal value?

- Set a variable named “previous_best_score” to store the best value from the previous generation
- The role of variables: measuring differences from previous generations
- Stop Criteria: Best value in previous generation - Best value in current generation < 0.000001

CONSTRAINT



- Contraind data: 1.8931645870650984 , -4.022838217869665 6

Q. How did you account for constraints in the genetic algorithm?

- Unconstraint genetic algorithm: The bounds parameter is used to define the constraint as the range of the variable.

- Constraint genetic algorithm: Using the is_feasible() function,
  Using the evolve() function

Q.Why constrained & unconstrained optimal points are different?

- Constraints: $x \leq 2$
    - The solution space is limited by constraints
- Hyperparameters: crossover rate, mutation rate of generations

UNITED PARCEL SERVICE (UPS)

- To find the shortest route, visiting each point once from origin (index 0 point)
- Choose 3 different cities, manage planning for drivers

Atlanta, GA

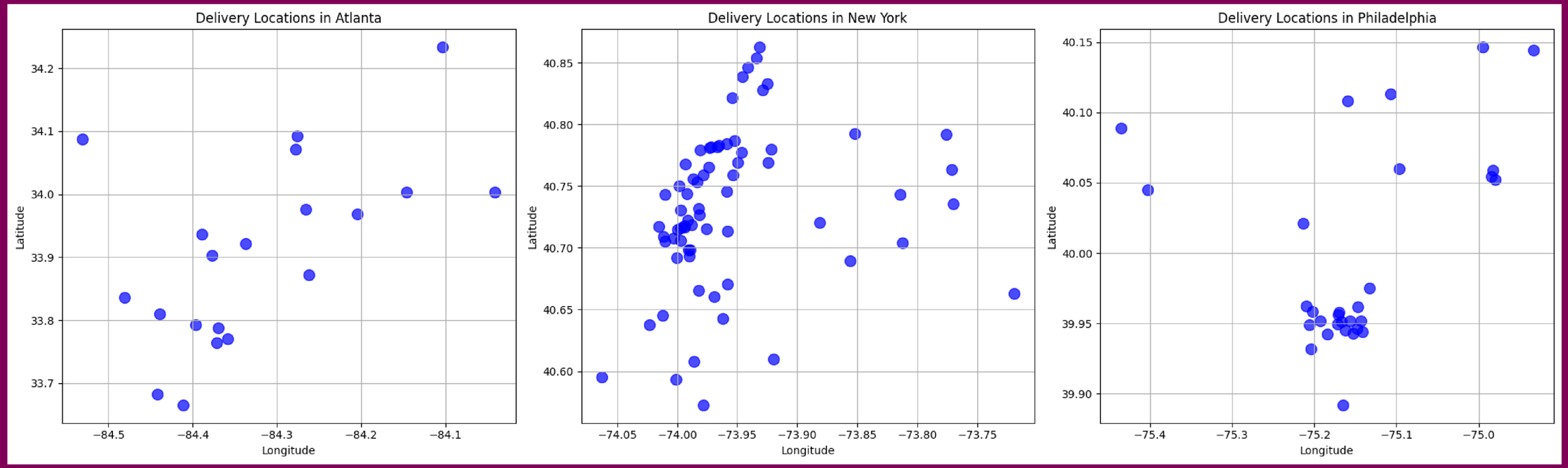
https://a.cdn-hotels.com/gdcs/production114/d1629/63a8dbe5-e678-4fe4-957a-ad367913a3fa.jpg

New York, NY

https://cdn.britannica.com/61/93061-050-99147DCE/Statue-of-Liberty-Island-New-York-Bay.jpg

Philadelphia, PA

https://cdn.britannica.com/42/94242-050-0BC6FC52/Philadelphia.jpg



Atlanta, GA

New York, NY

Philadelphia, PA

- **Implementing the Genetic Algorithm to Solve the TSP**

1) Initial setup : distance_matrix, other parameters

2) Creation of individuals and population : start = index 0

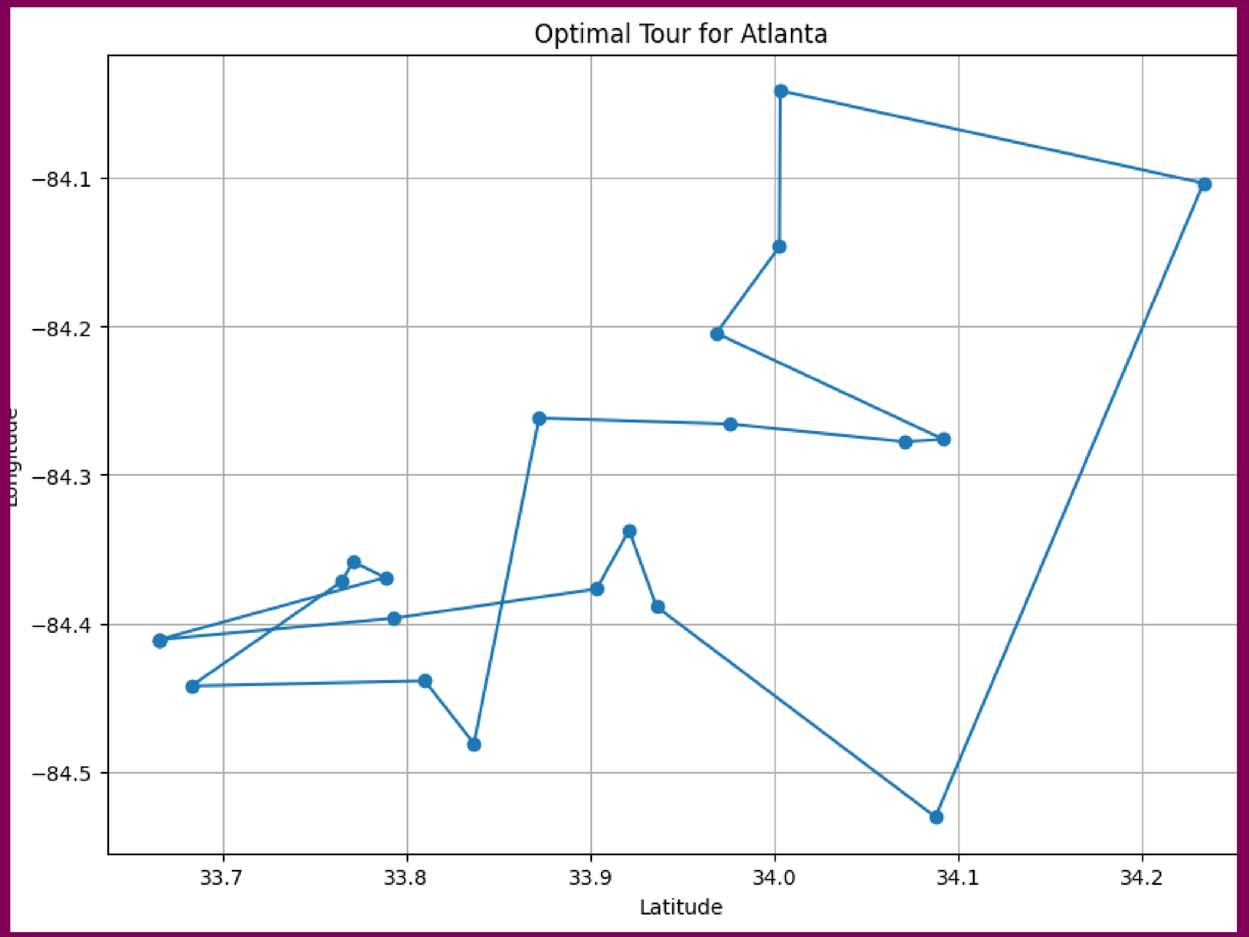
3) Define evaluation function : fitness of each individual

4) Register genetic operators : Crossover, mutation, selection (tournament=3)

5) Execute genetic algorithm

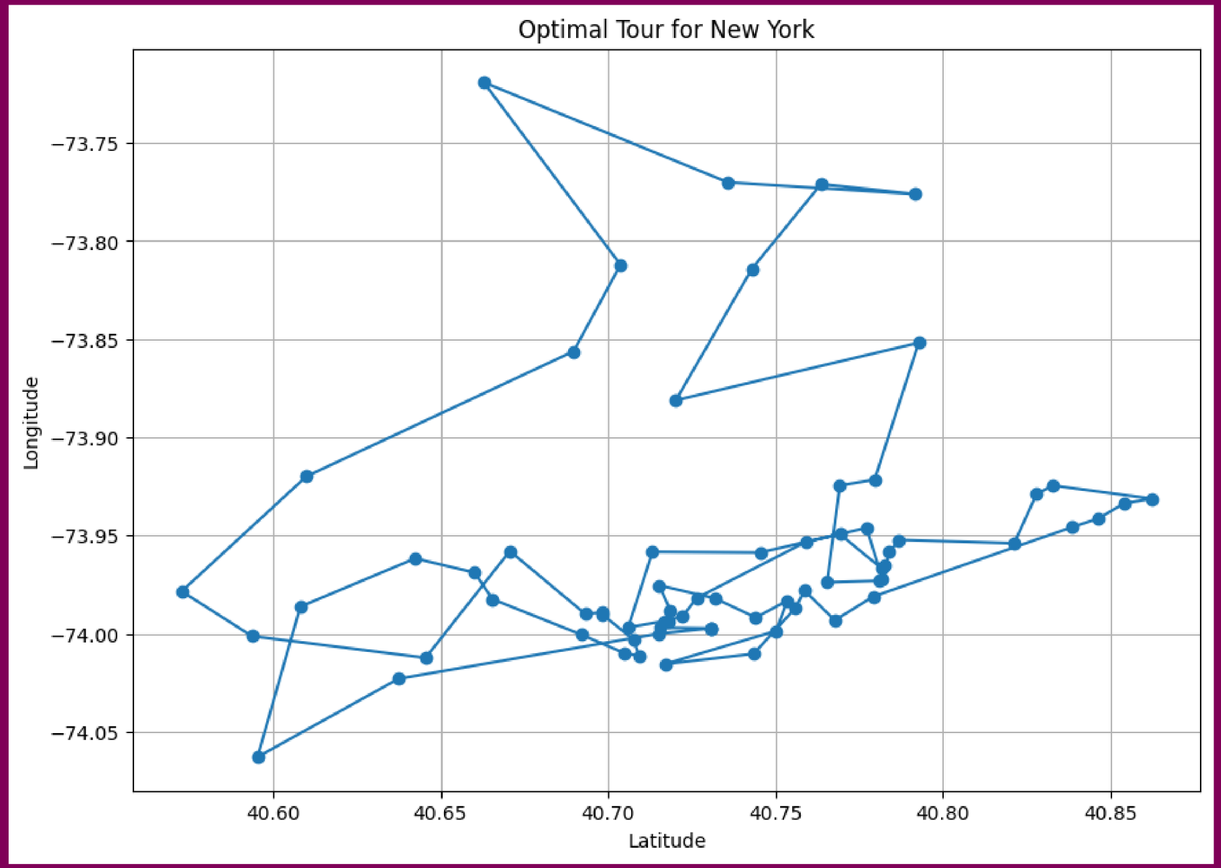
6) optimal solution : Select the best individual(shortest route)

- **Result of Atalnata**



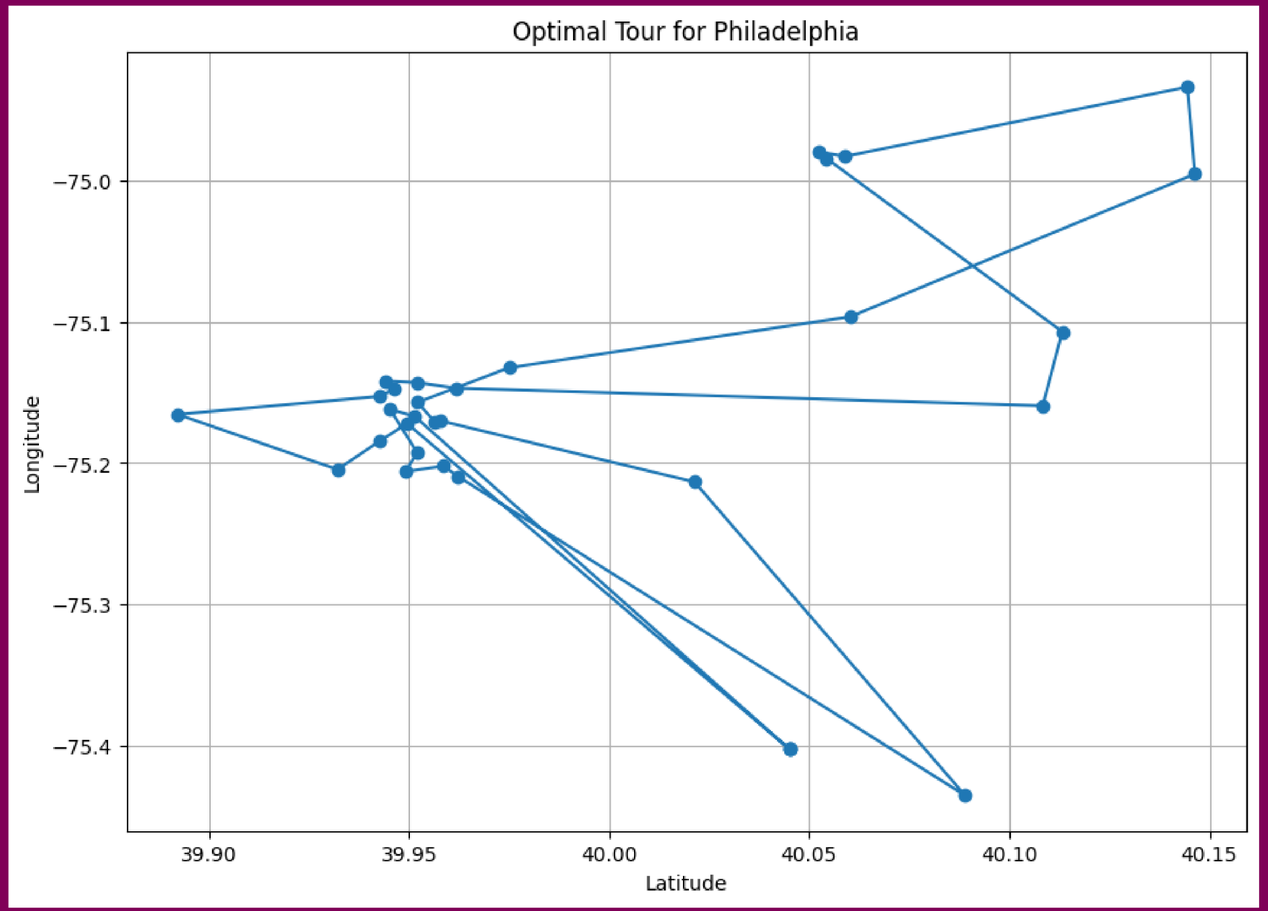
- Best tour in Atlanta: [0, 12, 10, 4, 19, 5, 3, 18, 14, 7, 15, 9, 13, 17, 11, 8, 6, 1, 2, 16, 0]
- Best distance in Atlanta: 231.28 km

Result of New York



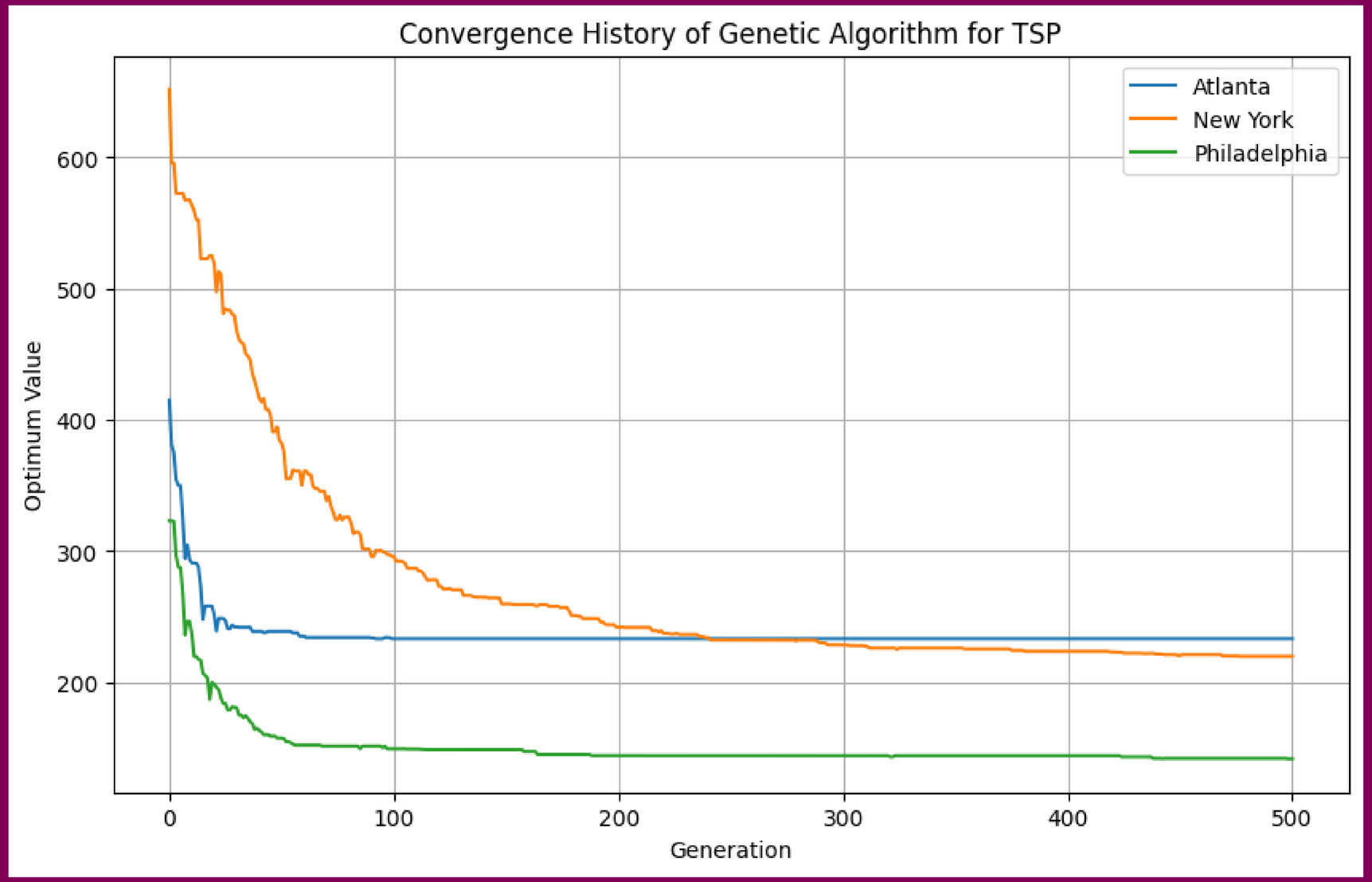
- Best tour in New York: [0, 4, 65, 25, 33, 6, 66, 63, 5, 22, 48, 59, 24, 29, 40, 37, 10, 54, 45, 42, 21, 51, 32, 56, 9, 17, 13, 30, 43, 49, 27, 1, 26, 16, 35, 15, 60, 11, 55, 28, 20, 62, 58, 8, 57, 14, 7, 67, 52, 36, 61, 38, 50, 34, 44, 64, 47, 19, 53, 18, 12, 3, 46, 23, 2, 41, 31, 39, 0]
- Best distance in New York: 203.15 km

Result of Philadelphia

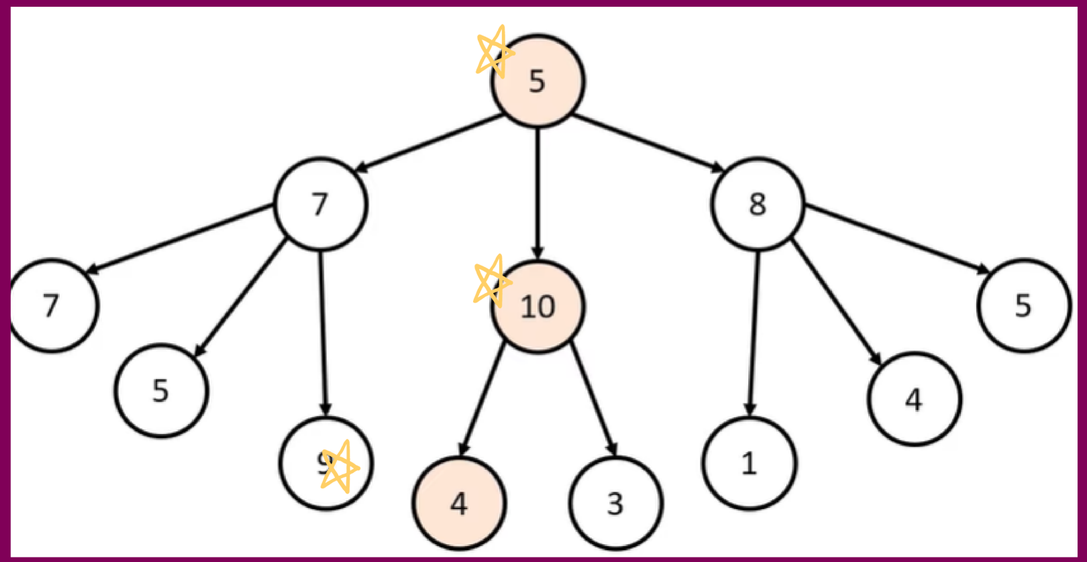


- Best tour in Philadelphia: [0, 4, 13, 15, 28, 27, 1, 29, 5, 14, 16, 9, 2, 6, 26, 7, 11, 22, 20, 8, 12, 24, 19, 23, 18, 21, 10, 25, 3, 17, 0]
- Best distance in Philadelphia: 155.72 km

- # Creat a plot of Convergence History



Greedy Algorithm



Concept

The steps in the Greedy algorithm mean that at each step you make choices that you think are optimal, until you finally find an overall optimal solution.

- **Implementing the Greedy Algorithm to Solve the TSP**

1) Initial setup : distance_matrix, other parameters

2) Creation of individuals and population : start = index 0

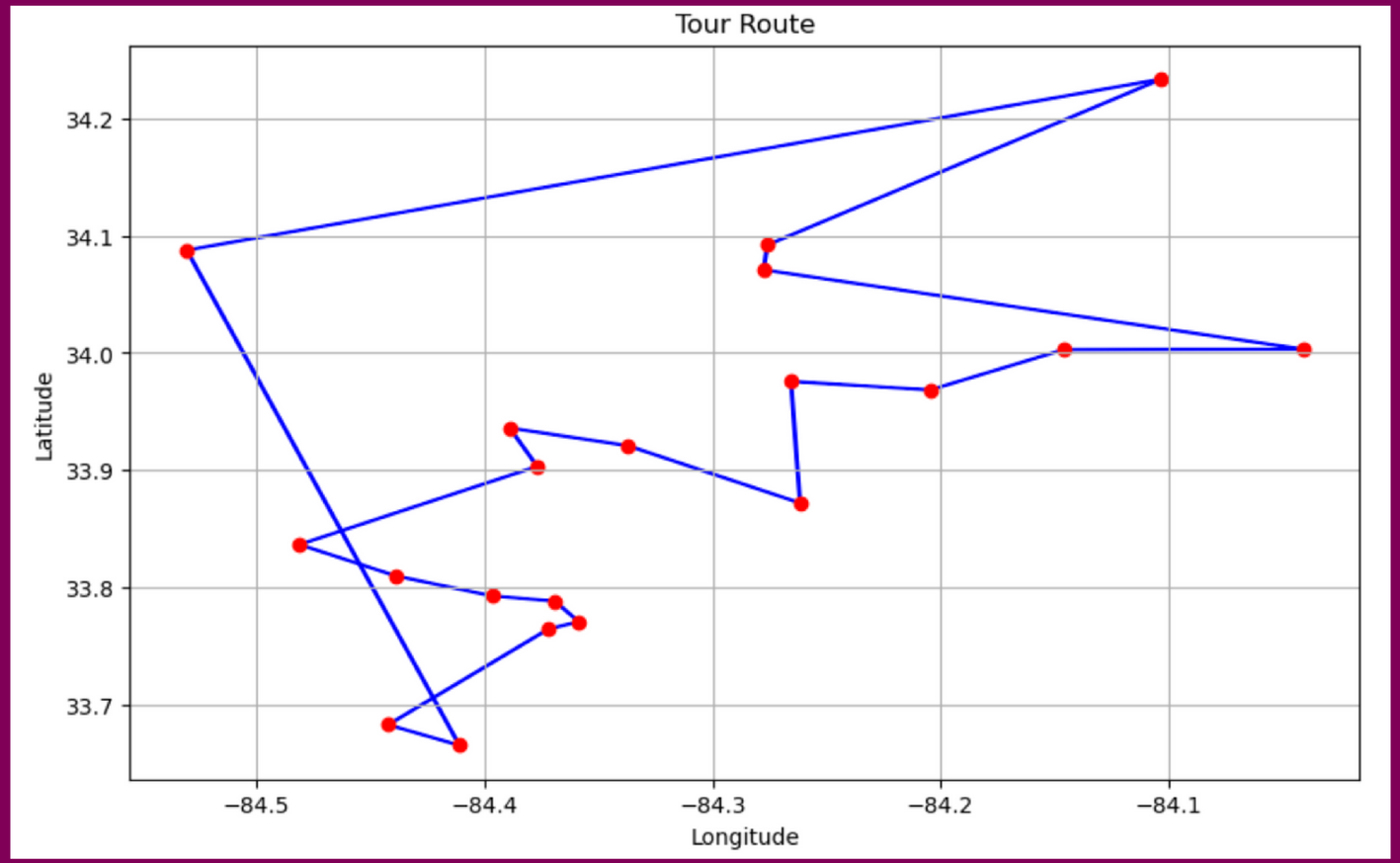
3) Repeat until all cities have been visited

4) Select the next city that is the shortest distance from each city

5) Return to the arrival point

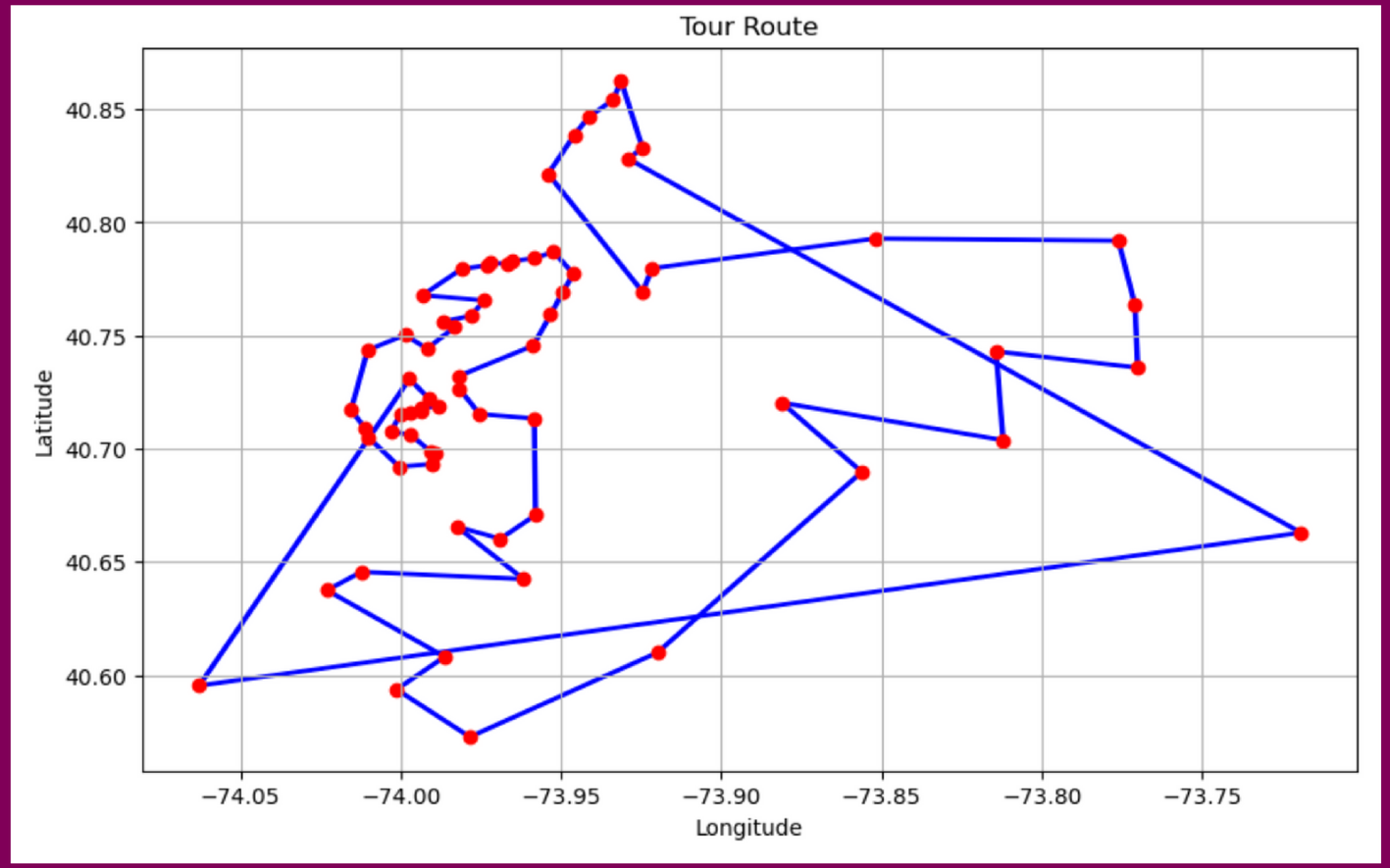
6) Add up all the paths travelled to come up with a solution

- **Result of Atalnata**



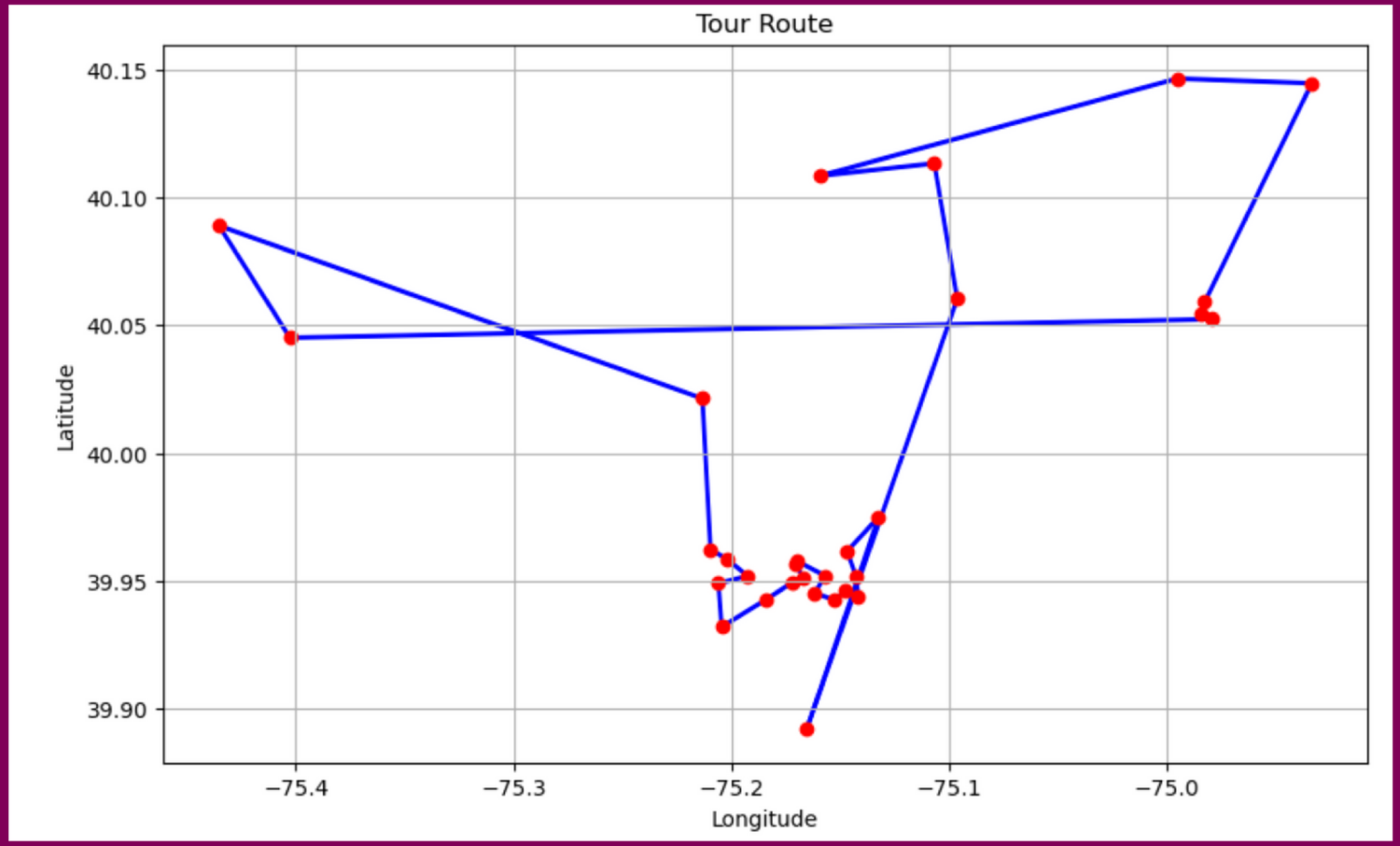
- Best tour in Atlanta: [0, 6, 1, 2, 16, 12, 8, 11, 10, 19, 4, 17, 13, 7, 14, 18, 9, 15, 3, 5, 0]
- Best distance in Atlanta: 231.98 km

- **Result of New York**



- Best tour in New York: [0, 11, 41, 31, 55, 39, 4, 59, 28, 24, 29, 40, 5, 22, 48, 18, 53, 12, 46, 3, 19, 47, 1, 64, 44, 26, 16, 8, 57, 14, 7, 35, 58, 15, 62, 23, 60, 2, 20, 37, 66, 63, 6, 10, 65, 33, 54, 45, 42, 21, 30, 51, 13, 56, 17, 9, 43, 49, 27, 67, 34, 50, 38, 61, 36, 52, 32, 25, 0]
- Best distance in New York: 202.55 km

- **Result of Philadelphia**



- Best tour in Philadelphia: [0, 23, 19, 18, 21, 25, 10, 15, 13, 4, 17, 12, 24, 8, 3, 27, 1, 29, 5, 14, 20, 28, 22, 9, 16, 11, 7, 26, 2, 6, 0]
- Best distance in Philadelphia: 155.12 km

Discussion

Q. How to improve Greedy algorithm efficiency

A . Using Different Methods

1. Finding the shortest distance

: Optimise the Greedy algorithm by directly selecting the minimum distance in the transformation matrix using Numpy's Find Minimum method.

2. Finding the city index value

: When moving on to the next city, I need a way to get the index value of that city and easily move to the next array.

Discussion

Q. How to improve Greedy algorithm efficiency

A . Transform DistanceAdjacencyMatrix

1. Matrix transformations

: It would have been easier to implement the algorithm if the distance transformation matrix was not a symmetric matrix, but a distance transformation matrix with duplicate values removed.

2. Converting to a dictionary value

: Simplify the Greedy algorithm by structuring the distances from city 0 to other cities into a nested list format resembling a JSON file, where city indices serve as keys and distances as corresponding values.

Q. Do we always get the same solution when executing the algorithm multiple times?

A. No. In each run, initialization of the population, genetic operators, and selection mechanisms induces randomness.

Q. Is solution from the genetic algorithm a global optimum?

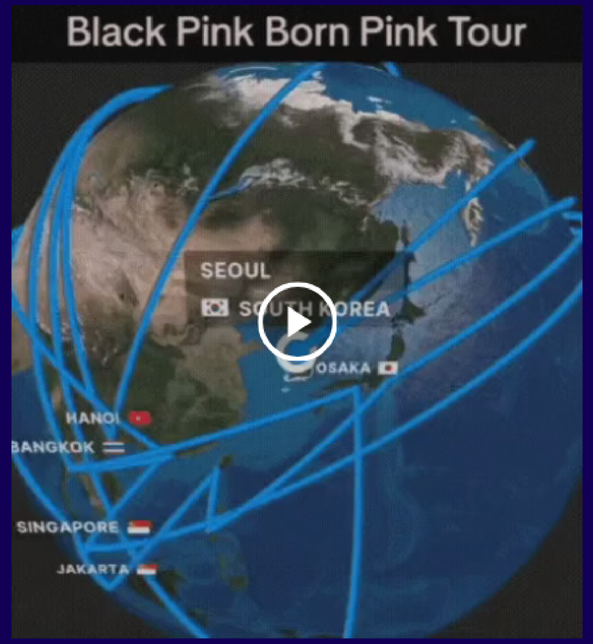
A. No. It is not guaranteed to be the global optimum. GA = diversity and efficiency. However, the quality depends on various factors. Although it gives the best possible solution in their trial, the best one is a feasible solution, not global!

Q. Can we improve the runtime performance of genetic algorithm?

A. Yes. Appropriate data structures and memoization to reuse results can do so.
To integrate greedy decision-making heuristics, choose the best and the fittest value.

FINDING THE BEST BLACKPINK WORLD TOUR SCHEDULE

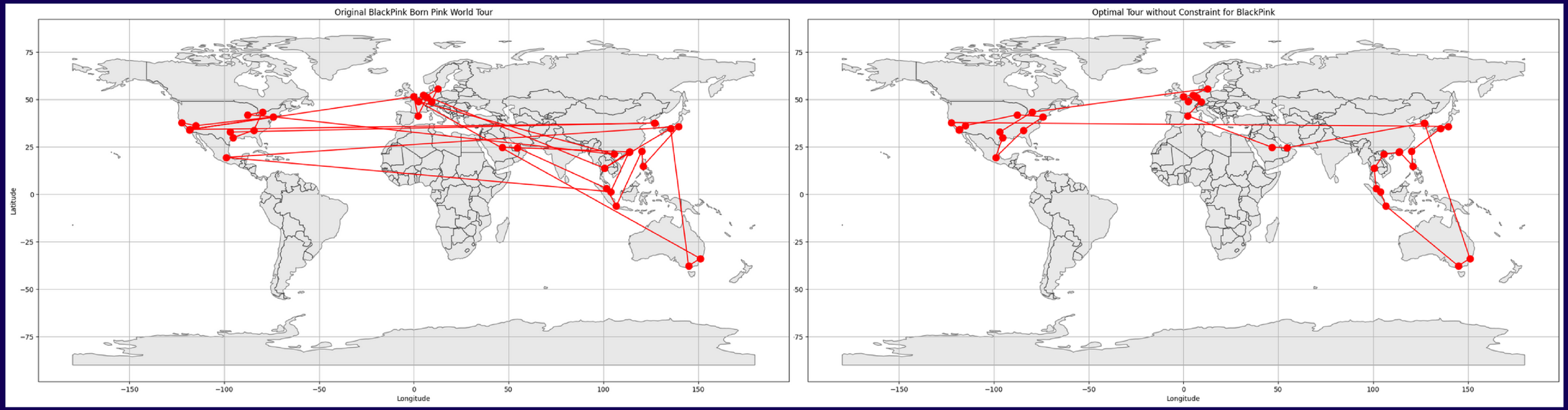
TOUR SCHEDULE



They travelled ~ **164K km** !!!

38 cities (including overlapping cities for the encore)

Distances without constraints



164,514.22 km ⟶ **63,957.53 km**

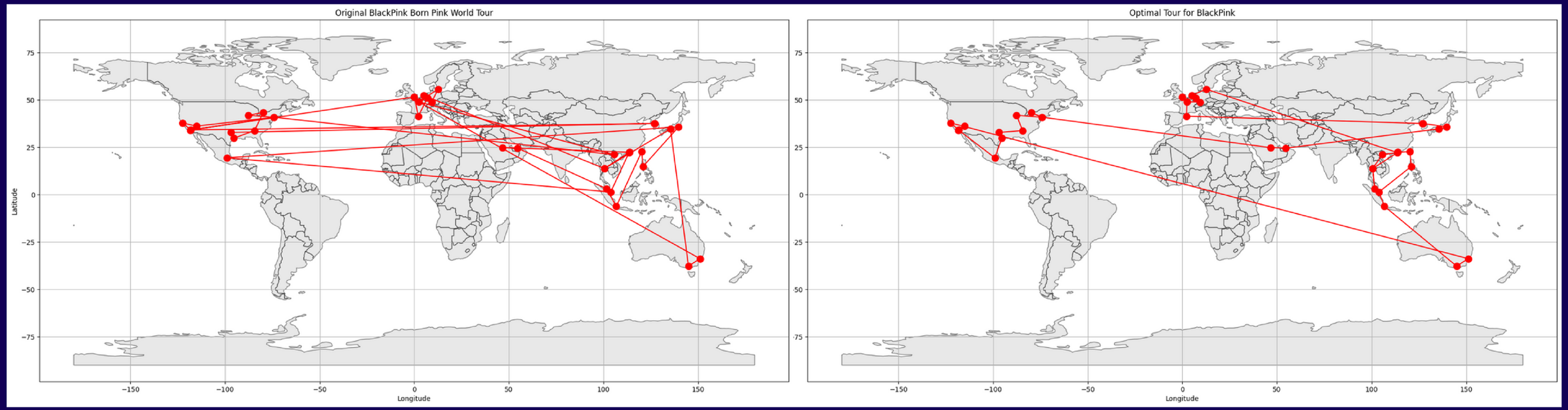
Reduced 100,556 km

TOUR SCHEDULE

- **Constraints:**
  - Revisiting for encore should not be connected directly to the first visit
  - Start & end in Seoul
- **Updates:**
  - **Additional parameters:** start_index, end_index, and min_distance.
  - **evalTSP function:** ensure the distance between cities > min_distance

Distances with constraints:

(1) Start pt = index 0, End pt = index 37
(2) Minimum distances between nodes = 500 km



164,514.22 km ⟶ **88,368.89 km**

Reduced 76,122 km


조선
민주주의
인민공화국
평양
고척스카이돔
올림픽공원 KSPO DOME
고척스카이돔
대한민국
황해

REFERENCE

- Anthropic. (2023). Claude. Claude.ai. https://claude.ai/chats
- Kim, J.H. (2024). Genetic Algorithm. Lecture Resources on Algorithm Class (April 26). Retrieved from https://lms.handong.edu/courses/15331/external_tools/3
- OpenAI. (2024). ChatGPT (April 26 version) [Large language model]. https://chat.openai.com

THANK YOU FOR LISTENING!

Don't hesitate to ask any questions!