Final Project Report

CSCI 490 - Spring 2025

Project Overview

- Generated a graph of *N* nodes, each node representing a gas station, and each edge representing an edge between them
- Defined three solutions to choosing a path through the graph, and two solutions to choosing a refueling policy for a given path
- Implemented the optimal solution, which chooses a path and finds the refueling policy at the same time
- Compare the accuracy and efficiency of all solutions

Constraints

Vehicle Constraints

- Tank capacity: max 20 gallons
- Miles per gallon: fixed at 20

Problem Constraints

- Fill the tank: At a given station, the only two options available are either to fill the tank to capacity or to move on to the next station.
- Scenario: The vehicle begins with a full tank at station 1, and the vehicle must refuel to a full tank at station 20

Station Network Constraints

- Station 1 = start, station 20 = destination
- Each station *i* has a random price normally distributed with mean $4.75 and standard deviation $0.20
- Each edge (*i*, *j*) has an attribute representing the distance from station *i* to station *j* that is:
  - Less than or equal to the max distance the vehicle can travel on a full tank
  - Greater than distance(i, j – 1)
  - Less than distance(i – 1, i) + distance(i – 1, j)

Pathfinding Solutions

Cheapest Station

1. Always advance to the next cheapest reachable station
2. Time efficiency: $O(n)$, space efficiency: $O(n)$

A* Search

1. Estimate cost to an adjacent station
2. Estimate cost from adjacent station to destination
3. Advance to the station that minimizes the sum of the above two estimations
4. Time efficiency: $O(n^2)$, space efficiency: $O(n^2)$

Floyd-Warshall

1. Get the shortest distance and next node matrices using the Floyd-Warshall algorithm
2. Follow the nodes in the next node matrix to find the shortest path from the start to the destination
3. Time efficiency: $O(n^3)$, space efficiency: $O(n^3)$

Refueling Solutions

Greedy

1. Find the farthest reachable station from the current station on a full tank
2. If the current station is the cheapest in range of a full tank, or there is not enough fuel in the tank to reach the cheapest station in range, fill the tank to capacity
3. Advance to the next station
4. Time efficiency: $O(n^2)$, space efficiency: $O(n)$

Dynamic

1. Starting from the destination, find the minimum cost to the destination
2. Minimum cost to the destination = cost to the next station + minimum cost from the next station to the destination
3. Go back to the previous station
4. Time efficiency: $O(2^n)$, space efficiency: $O(n)$

Optimal Solution

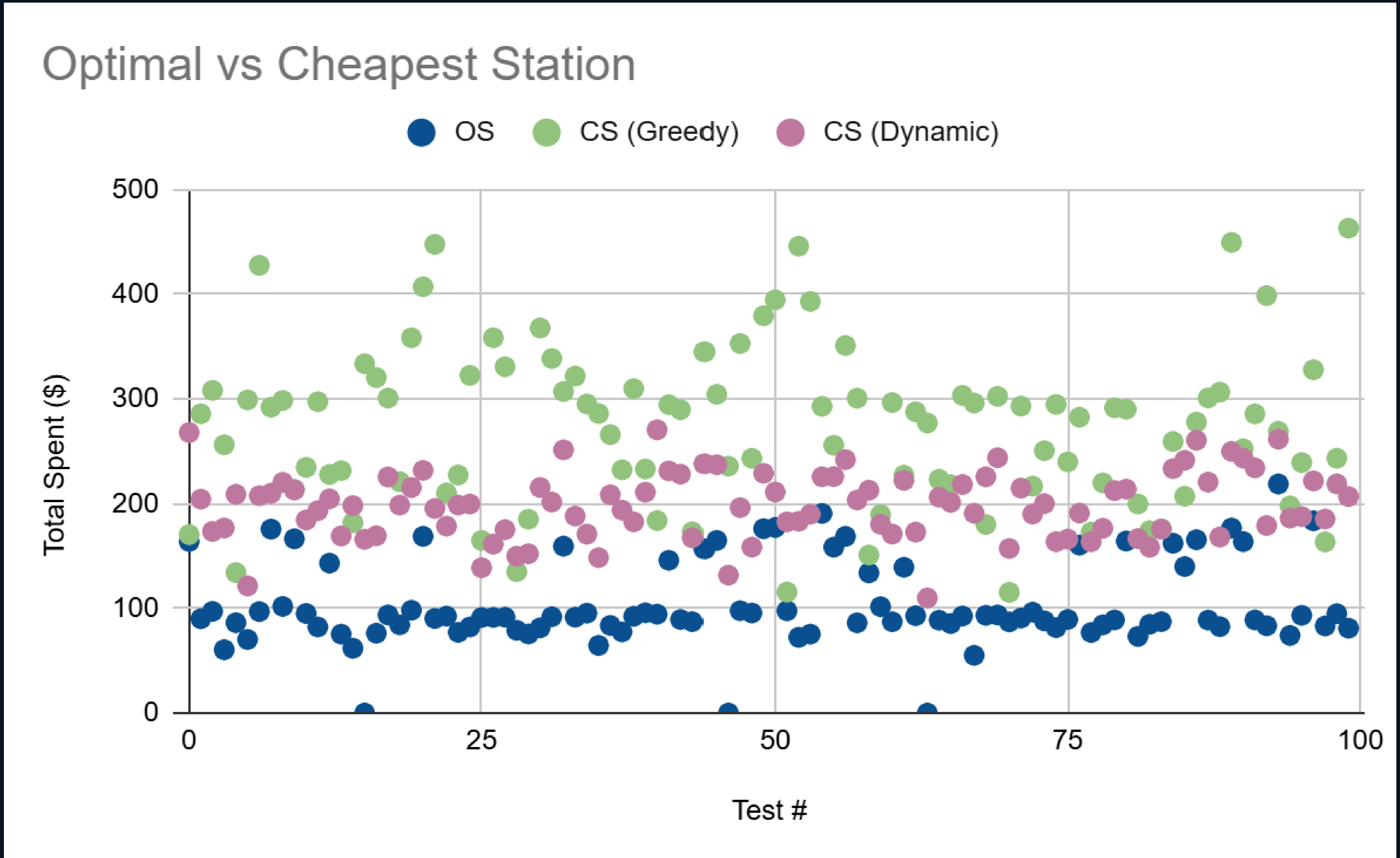
1. Get shortest distance and next node matrices using the Floyd-Warshall algorithm
2. Follow the nodes in the next node matrix to find the shortest path from the start to the destination
3. At each station on the path, if there is not enough fuel to reach the next station, fill the tank to capacity
4. Time efficiency: $O(n^3)$ (really $n^3 + n$ for the Floyd-Warshall algorithm followed by tracing the path in the next node matrix), space efficiency: $O(n^2)$ (more specifically, $n + 3n^2$)

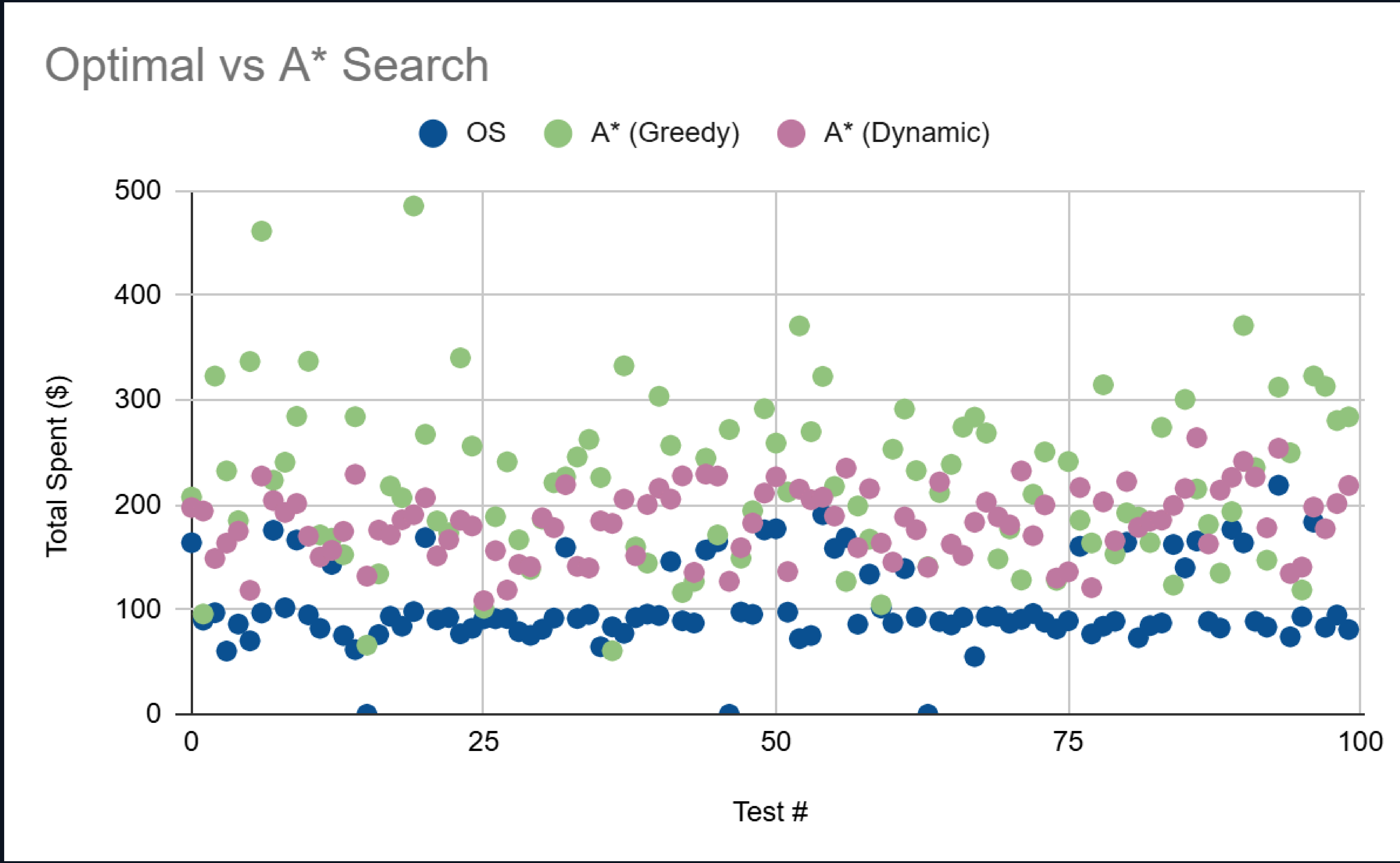
Collecting Observational Data

For 100 randomly generated routes:

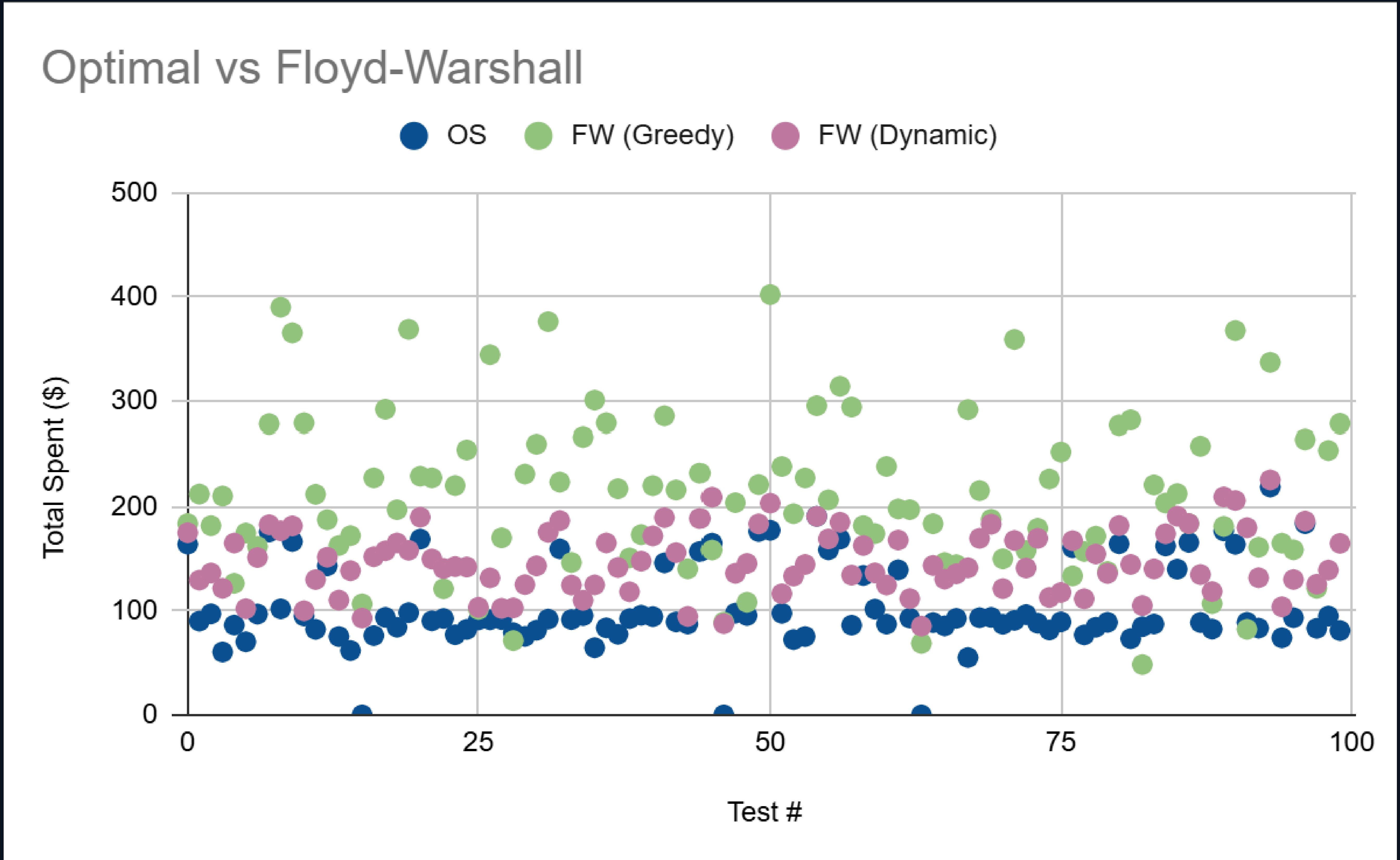
1. Apply the optimal solution to find the best total spent for the route
2. Apply the Cheapest Station, A* Search, and Floyd-Warshall solutions for choosing a path through the graph
3. For each path obtained in step 2, apply the greedy and dynamic solutions to find two totals spent for each path
4. Record and analyze the results in a dataframe and .csv file

Results


Optimal vs Cheapest Station
OS
CS (Greedy)
CS (Dynamic)
Total Spent ($)
500
400
300
200
100
0
0
25
50
75
100
Test #



Optimal vs A* Search
OS
A* (Greedy)
A* (Dynamic)
500
400
300
200
100
0
Total Spent ($)
0
25
50
75
100
Test #

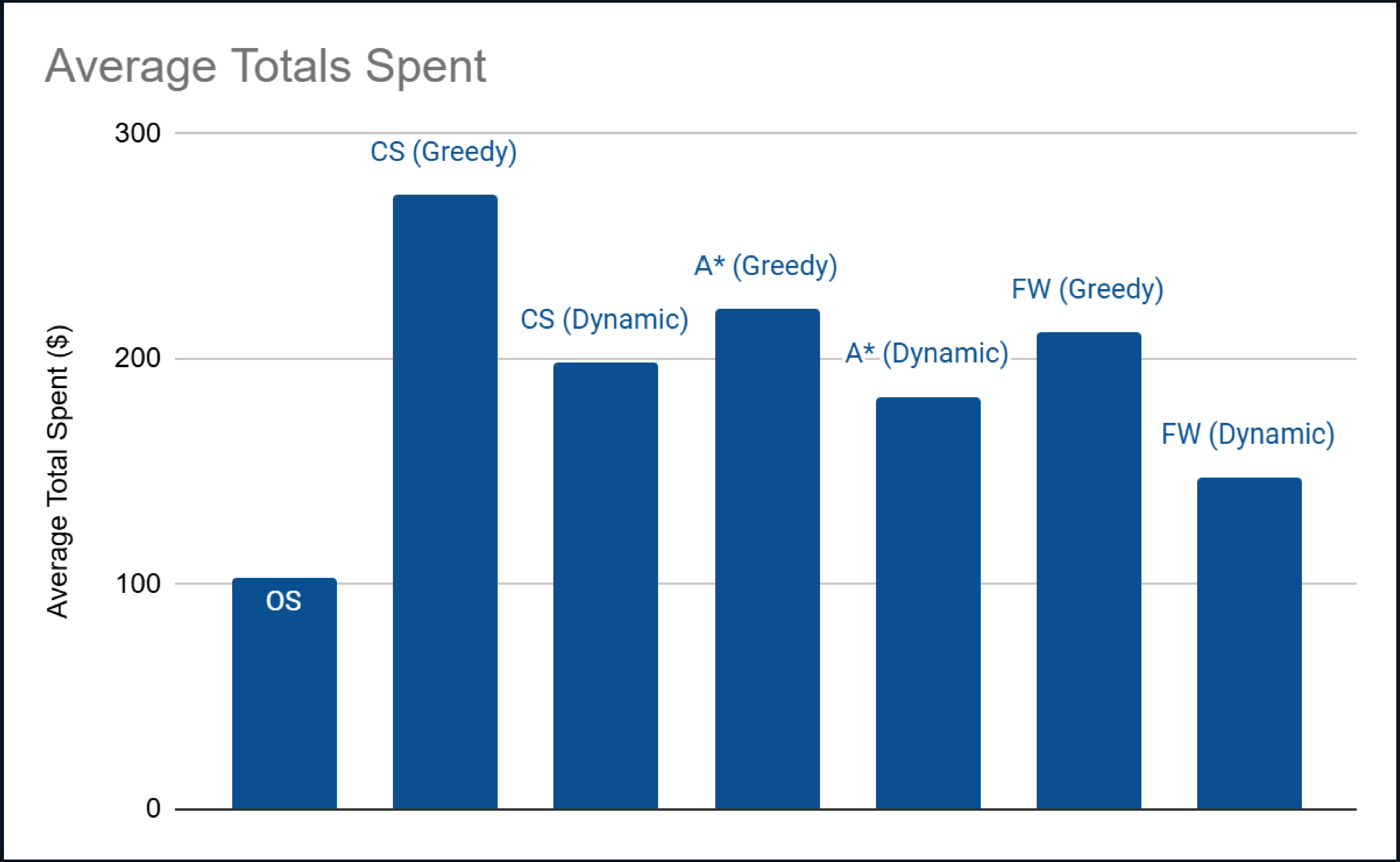


Optimal vs Floyd-Warshall
OS
FW (Greedy)
FW (Dynamic)
Total Spent ($)
0
100
200
300
400
500
0
25
50
75
100
Test #

Accuracy Compared to Optimal Solution

$ (%) more expensive than optimal solution on average:

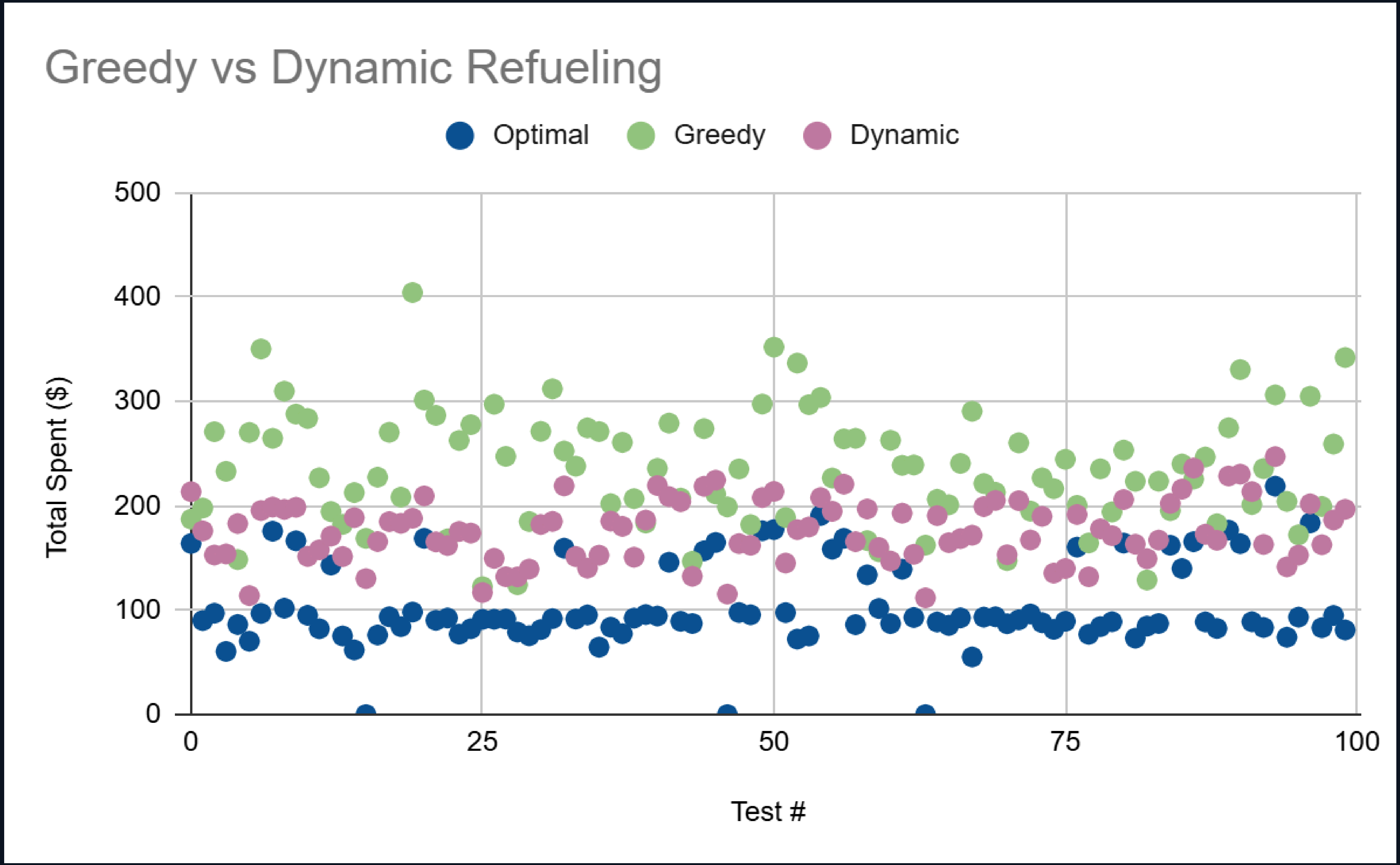
- Cheapest Station
  - Greedy refueling: $169.84 (62.2%)
  - Dynamic refueling: $95.15 (47.96%)
- A* Search
  - Greedy refueling: $118.95 (53.54%)
  - Dynamic refueling: $80.02 (43.67%)
- Floyd-Warshall
  - Greedy refueling: $108.91 (51.34%)
  - Dynamic refueling: $44.21 (29.98%)



Average Totals Spent
Average Total Spent ($)
300
200
100
0
OS
CS (Greedy)
CS (Dynamic)
A* (Greedy)
A* (Dynamic)
FW (Greedy)
FW (Dynamic)

Greedy vs Dynamic Refueling Accuracy

Greedy refueling is, on average, 25.21% ($59.44) more expensive than dynamic refueling



Greedy vs Dynamic Refueling
Optimal
Greedy
Dynamic
500
400
300
200
100
0
Total Spent ($)
0
25
50
75
100
Test #

Discussion

- Greedy refueling is moderately more expensive, but also moderately more efficient, than dynamic refueling. Optimal refueling is both less expensive and more efficient.

- Floyd-Warshall solution is the closest in total spent to the optimal solution, but only using dynamic refueling.

- Under the fill-the-tank constraints, the overall best solution (optimal solution) is also the most intuitive: Choose the shortest path by distance and fill the tank only when you must refuel to reach the next station.

Next Steps

- How would the results change if the fill-the-tank constraints were lifted?
- How do the standard problem results change compared to one with a maximum number of refueling stop constraints?