A Novel Approach to Route Similarity Measures for Shared Mobility Matching Systems

Antoniu Negrea

2025

Contents

Chapter 1

Modeling the Experimental Part

This chapter rigorously describes the data used, the planned experiments, the mathematical modeling of the similarity measures, the algorithms compared, and the validation methods. The goal is to demonstrably prove, in a reproducible and analytically supported manner, that the proposed approach brings improvements over existing methods found in the literature.

1.1 The Dataset

1.1.1 Representation of Urban Routes

Each route is modeled as an ordered list of GPS points:

$$R = \{(lat_1, lon_1), \ldots, (lat_n, lon_n)\}.$$

The data sources are:

- simplified urban road network
- artificially simulated routes for controlled scenarios

To reduce complexity, the GPS points are projected onto a discretized network $G = (V, E)$, where V are intersections and E are segments.

1.1.2 User Profiles

Each user is associated with a triplet:

$$U_i = (o_i, d_i, t_i),$$

where o_i is the origin, d_i the destination, and t_i the temporal interval (time window).

1.2 Simplified Similarity Measures

The proposed methodology uses two measures that are easy to implement:

1.2.1 Geometric Similarity

For two routes R_1, R_2:

$$S_{geo}(R_1, R_2) = 1 - \frac{1}{|R_1|} \sum_{p \in R_1} \min_{q \in R_2} d(p, q),$$

where $d(p, q)$ is the Haversine distance.

1.2.2 Segment Overlap

$$S_{overlap}(R_1, R_2) = \frac{|R_1 \cap R_2|}{\max(|R_1|, |R_2|)}.$$

1.2.3 The Final Similarity Function

A simple linear function:

$$S_{final}(R_1, R_2) = \alpha S_{geo}(R_1, R_2) + \beta S_{overlap}(R_1, R_2),$$

where $\alpha + \beta = 1$.

1.3 Matching Algorithms

1.3.1 Static Matching — Partition Merging

The objective is to group users such that the cost is minimized:

$$\text{cost}(G) = \sum_{i,j \in G} (1 - S_{final}(R_i, R_j)).$$

1.3.2 Dynamic Matching — Greedy

For a new request:

$$\Delta\text{cost} = \text{cost}(G \cup \{U_k\}) - \text{cost}(G).$$

The request is allocated to the group with the minimum Δcost.

1.4 Proposed Experiments

1.4.1 Experiment 1: Static vs. Greedy

The comparison between the two algorithms (Static Partition Merging and Dynamic Greedy) uses two principal metrics:

1. **Efficiency** (Travel Gain): The reduction in total travel distance.

$$\text{TravelGain} = \frac{D_{solo} - D_{shared}}{D_{solo}}.$$

2. **Equity** (Maximum Relative Detour, MRD): The highest proportional increase in travel distance/time experienced by any single rider in a shared group. This assesses the fairness of the solution.

1.4.2 Experiment 2: Impact of Parameters α, β

The influence of weights on matching quality is analyzed using both the **Travel Gain** and **MRD** metrics to assess the trade-off between efficiency and equity.

1.4.3 Experiment 3: Scalability

We measure:

- runtime;
- memory used.

1.5 Validation Methods

Validation is exclusively numerical:

1.5.1 Internal Validation

Repeated simulations with artificially generated data.

1.5.2 External Validation

Comparison of results with:

- Xia & Curtin (2019) – spatial model;
- Duan (2018) – partition merging;
- Sun (2023) – greedy.

1.6 Conclusion

The experimental model is simplified, reproducible, and easy to implement. It allows for the evaluation of similarity functions and matching algorithms.

Chapter 2

Case Study on the Initial Dataset

This chapter presents a controlled experiment performed on a small simulated dataset to validate the proposed methodology in a simple scenario.

2.1 Dataset Description

The initial set contains:

- 10 short routes with controlled characteristics (50–200 m);
- close or distant origins and destinations;
- simple intersections for ease of implementation.

Three types of scenarios are included, each run with 10 routes:

1. nearly identical routes (IDENTICAL);
2. partially overlapping routes (PARTIAL);
3. completely different routes (DIFFERENT).

2.2 Experimental Code Implementation

The practical component consists of implementing the following in Python:

- a route generator;
- the S_{geo} and $S_{overlap}$ functions;
- the static and greedy algorithms;
- the metrics measurement module.

Code structure:

```
ExperimentalPart/
    route_generator.py
    similarity.py
    static_matching.py
    greedy_matching.py
    metrics.py
    main.py
```

2.3 Results and Analysis

The initial simulations were executed using $N = 10$ routes per scenario, comparing the Static Matching (Partition Merging) against the Dynamic Matching (Greedy) approach. The results are summarized below.

2.3.1 Experiment 1: Static vs. Greedy ($\alpha = \beta = 0.5$)

The initial experiment focuses on baseline performance using balanced similarity weights.

- **Identical Scenario:** As expected, both algorithms performed optimally, merging all 10 routes into a single group, yielding the maximum possible 90.00% Travel Gain and zero detour (0.00% MRD). This validates the algorithms' ability to identify perfect matches.
- **Partial Scenario:** The **Static Matching** algorithm achieved a slightly higher Travel Gain (18.87%) compared to the Greedy approach (15.09%), indicating that its global optimization view resulted in marginally more efficient overall groupings, even though both formed the same number of groups (6) and had the same average size (1.67) and MRD (50.00%).

Table 2.1: Experiment 1: Static vs. Greedy Comparison ($\alpha = 0.5, \beta = 0.5$)

Scenario	Algorithm	Groups	Avg. Size	Travel Gain	MRD
IDENTICAL	Static	1	10.00	90.00%	0.00%
	Greedy	1	10.00	90.00%	0.00%
PARTIAL	Static	6	1.67	18.87%	50.00%
	Greedy	6	1.67	15.09%	50.00%
DIFFERENT	Static	8	1.25	7.69%	100.00%
	Greedy	6	1.67	15.38%	100.00%

- **Different Scenario:** The results here are highly revealing. While both algorithms correctly showed low efficiency and high detours (100.00% MRD), the Greedy algorithm unexpectedly yielded a better Travel Gain (15.38% vs. Static's 7.69%) despite forming fewer groups (6 vs. 8). This suggests that in scenarios with poor inherent matchability, the sequential nature of the Greedy algorithm might, by chance, establish a few highly efficient initial groups that the Static algorithm's global, similarity-driven cost function failed to identify under this specific weight setting.

2.3.2 Experiment 2: Impact of Parameters (α, β) on 'Partial' Scenario

This experiment used the PARTIAL scenario as a testbed to analyze how the relative weighting of geometric similarity (α) versus segment overlap (β) affects efficiency and equity.

- **Geometric-Heavy ($\alpha = 0.9, \beta = 0.1$):** This setting proved to be the most successful for maximizing efficiency, with the **Greedy algorithm achieving the highest Travel Gain (30.91%)** across all tests. This result confirms that S_{geo} (geometric proximity) is the most informative metric in our similarity function. However, the Static algorithm achieved a high gain (29.09%) while maintaining a significantly lower detour (**MRD 20.00%** vs. Greedy's 50.00%), highlighting the trade-off: Static matching offers superior equity for similar efficiency.

Table 2.2: Experiment 2: Parameter Impact on Matching Quality (PARTIAL Scenario)

α	β	**Algorithm**	**Travel Gain**	**MRD**
0.1 (Overlap-Heavy)	0.9	Static	25.45%	0.00%
		Greedy	25.45%	0.00%
0.5 (Balanced)	0.5	Static	29.09%	20.00%
		Greedy	20.00%	50.00%
0.9 (Geometric-Heavy)	0.1	Static	29.09%	20.00%
		Greedy	**30.91%**	50.00%

- **Overlap-Heavy ($\alpha = 0.1, \beta = 0.9$):** This setting resulted in perfect equity (0.00% MRD) for both algorithms, but at the cost of constrained efficiency (25.45% gain). This high β weight makes the similarity function overly strict, only permitting near-identical route matches, which limits the potential for efficiency gains by excluding feasible, but slightly detoured, matches.

2.4 Conclusion

The initial set **quantitatively confirms** that the two simple similarity measures are sufficient for relevant and measurable experiments. The results validate the model by demonstrating clear performance differences—for instance, the dependence on parameter weighting and the inherent trade-off between the Static (equity-focused) and Greedy (efficiency-focused) algorithms. The experiments successfully demonstrated the sensitivity and impact of the final similarity function's weighting, with the S_{geo} component proving to be a critical factor for high-quality matching.