Via Data Challenge

Patrick Nevill,patricklenevill@gmail.com

> **Question 1**
>
> Propose a metric and/or algorithm to assess the potential efficiency of aggregating rides from many vehicles into one microtransit van/bus, given the available data. Make realistic assumptions and any necessary simplifications and state them.

Metric Choice

Microtransit services achieve efficiency by aggregating multiple passenger trips into shared vehicles. However, not all travel demand patterns are equally suitable for aggregation - success depends on whether trips have compatible origins, destinations, and timing. This analysis addresses the question: How can we quantify the efficiency potential of ride aggregation for a given set of trips? We propose Vehicle Miles Traveled (VMT) Savings as a metric that measures the theoretical reduction in total vehicle distance when trips are optimally matched and routed, subject to realistic quality-of-service (QoS) and capacity constraints.

VMT reduction has been chosen because it is the fundamental mechanism through which microtransit achieves its primary objectives: lower fuel consumption, reduced driver time, decreased operational costs and fewer vehicles on the road. Higher VMT savings indicate more efficient spatial-temporal matching of passengers, demonstrating strong aggregation potential.

$$\text{VMT Savings}(\Delta, C) = 1 - \frac{\text{VMT}_{\text{microtransit}}}{\text{VMT}_{\text{base}}} \tag{1}$$

Where:

- $\mathbf{VMT_{baseline}}$: Total vehicle miles required to serve the same set of requests with solo trips (baseline taxi operation), computed as the sum of shortest-path distances for each trip traveling alone.
- $\mathbf{VMT_{microtransit}}$: Total vehicle miles required by the proposed microtransit service to serve the same requests using shared routes, respecting capacity (C) and maximum delay (Δ) constraints.

VMT Scope: Both VMT baseline and VMT microtransit only include passenger-carrying travel distances. The dead mileage can be a significant proportion with Cramer and Krueger (2016) estimating the percentage of hours that taxis in New York are not carrying passengers is 51.7%. Understanding dead mileage changes would require an understanding of how taxi behavior would respond to transit changes which has not been done in this work.

Algorithm

Overview

End-to-end optimization of microtransit with simultaneous aggregation of many trips is a variant of the Dial-a-Ride Problem (Paepe et al. (2004)). The task is to assign every trip request to exactly one vehicle and determine routes for the vehicles under capacity and QoS constraints. With the goal to minimize the total travel distance of all the vehicles. This is an NP-hard problem and although algorithms do exist to solve this Alonso-Mora et al. (2017), the complexity of them at a city scale would be unfeasible for this scope of work.

The algorithm I propose is based on Santi et al. (2014), which creates a pairwise shareability-graph with maximum-weight matching, therefore it captures a large share of potential VMT savings by matching trip pairs and does not require solving the vehicle allocation problem. This approach could be extended to > 2 trip aggregation which leads to a hypergraph formulation, however this becomes computationally infeasible at the city scale.

Geographic pruning is added to reduce the computations required to produce the shareability graph and then large components of shareability graph are then partitioned to speed the computation of the maximum-weight matching.

Assumptions and Simplifications

- **Same requests:** The same set of requests are served in both baseline and microtransit scenarios.
  - In real life, we expect peoples travel behavior to respond to changes in quality of service. ie. Trip requests may increase if travel is less expensive.
- **Complete Demand Knowledge:** Full knowledge of future request demand is known.
  - Complete knowledge of future demand is unrealistic and in a practical online algorithm it would have to respond to current demand. Furthermore, our comparison is with previous taxi service which did not have this future knowledge.
- **Constant travel speed:** Constant travel speed has been set at 30km/h.
  - Speeds vary drastically with both time and location. Both the taxi trips and the new shared trips have been calculated with this speed so it will not affect the VMT, however it will mean that the calculated travel delay will vary in real life.
- **Unconstrained fleet size:** Every route proposed is able to be served by the microtransit fleet.
  - Without solving the vehicle allocation problem it is unknown what the vehicle requirements would be for the microtransit service.
- **Pick-up and drop-off from road intersections:** Passengers in all new routes will be picked-up and dropped-off from road intersections
  - Manhattan blocks are fairly small so the walk would be < 100m.

- **Instant pick-up and drop-off:** The algorithm does not include any additional time for picking up and dropping off passengers along the route.
  - The vehicle stopping, passengers finding the vehicle and boarding would all introduce further delays on a route.

Mathematical Problem Statement

Road network and travel model. Let $G = (V, E_G)$ be a directed road graph. Let $D : V \times V \to \mathbb{R}_+$ denote the shortest-path *distance* (meters) and $TT : V \times V \to \mathbb{R}_+$ the corresponding shortest-path *travel time* (seconds). These are computed on G and cached for all trip nodes used.

Trips and constraints. Let T be the set of trip requests. Each trip $i \in T$ is a tuple (o_i, d_i, s_i, p_i), where $o_i, d_i \in V$ are origin and destination nodes, s_i is pickup time (seconds from epoch), and p_i is passenger count. The service imposes:

- **Capacity** C: the vehicle load $\leq C$ at all times.
- **Maximum delay** Δ: for each passenger i, dropoff time under sharing $\mathrm{dt}_i^{\mathrm{shared}}$ must satisfy $\mathrm{dt}_i^{\mathrm{shared}} - \mathrm{at}_i^{\mathrm{solo}} \leq \Delta$, where $\mathrm{at}_i^{\mathrm{solo}} = s_i + TT(o_i, d_i)$.

Pairwise shareability and objective. For any pair (i, j), consider the four service orders $r \in \{o_i \to o_j \to d_i \to d_j,\ o_i \to o_j \to d_j \to d_i,\ o_j \to o_i \to d_i \to d_j,\ o_j \to o_i \to d_j \to d_i\}$. A route r is feasible if there exist pickup times consistent with s_i, s_j such that (i) vehicle load constraints hold and (ii) both passengers' added delays are $\leq \Delta$.

Define the shareability graph $S = (T, E_S)$ where an edge $(i, j) \in E_S$ exists if at least one route r for the pair is feasible. Let solo distances $d_i = D(o_i, d_i)$ and $d_j = D(o_j, d_j)$. For any feasible route r, define the pairwise VMT savings

$$w_{ij}(r) = d_i + d_j - D_{\mathrm{shared}}(r),$$

and set $w_{ij}^{max} = \max_{r \in \mathcal{R}_{ij}} w_{ij}(r)$. We then seek a maximum-weight matching on S:

$$\max \sum_{(i,j) \in E_S} w_{ij}^{max} x_{ij} \quad \text{s.t.} \quad \sum_{j:(i,j) \in E_S} x_{ij} \leq 1 \ \ \forall i \in T,\ x_{ij} \in \{0, 1\}.$$

This yields an optimal set of disjoint pairs under the pairwise model. The implied system-level VMT savings is

$$\text{VMT Savings}(\Delta, C) \ = \ \frac{\sum_{i \in T} d_i \ - \ \Big(\sum_{(i,j) \in M} D_{\mathrm{shared}}(i, j) \ + \ \sum_{k \in U} d_k \Big)}{\sum_{i \in T} d_i}, \tag{2}$$

where M is the set of matched pairs and U are unmatched trips served solo.

Algorithmic Walkthrough

1. **Data loading and filtering.** Read trips in daily chunks; apply Manhattan-boundary (polygon); retain (o_i, d_i, s_i, p_i).

2. **Map trips to road nodes.** Use vectorized nearest-node lookup to map lat/lon to nodes (`osmnx`) $o_i, d_i \in V$; .

3. **Precompute distances/times on trip nodes.** Build a persistent cache for $D(u, v)$ and $TT(u, v)$ over the set of unique trip nodes for O(1) lookups.

4. **Compute actual trip times.** Compute $\mathrm{at}_i^{\mathrm{solo}} = s_i + TT(o_i, d_i)$.

5. ***Vectorized pruning.** Generate candidate pairs via array-based filters: temporal overlap, capacity feasibility ($p_i + p_j \leq C$), and 4-point geographic proximity on (o_i, o_j, d_i, d_j).

6. **Feasibility checks.** For each candidate (i, j), evaluate the four service orders. Using pre-computed $TT(\cdot, \cdot)$, derive feasible pickup-time intervals and compute shared dropoff times $(\mathrm{dt}_i, \mathrm{dt}_j)$. Accept the route if $\mathrm{dt}_i - \mathrm{at}_i^{\mathrm{solo}} \leq \Delta$ and $\mathrm{dt}_j - \mathrm{at}_j^{\mathrm{solo}} \leq \Delta$.

7. **Edge construction.** For feasible (i, j), set $w_{ij} = d_i + d_j - D_{\mathrm{shared}}$ using $D(\cdot, \cdot)$; add edge to S with auxiliary analytics (delays, efficiency, savings%).

8. ***Shareability Graph Partitioning.** For feasibility, the shareability graph is partitioned by connected components; very large components are subdivided using Louvain community detection, with a target of ∼10k nodes/partition.

9. **Maximum-weight matching.** Each partition $G_k = (V_k, E_k)$ is matched independently using a maximum-weight matching (`rustworkx`).

*Heuristics

To improve scalability, I introduce geographic pruning heuristics that may eliminate some optimal pairings in the shareability graph.

$$\|\mathbf{loc}(o_i) - \mathbf{loc}(o_j)\|^2 \leq (D_{\mathrm{geo}})^2 \quad \text{AND} \quad \|\mathbf{loc}(d_i) - \mathbf{loc}(d_j)\|^2 \leq (D_{\mathrm{geo}})^2$$

Where $D_{\mathrm{geo}} = 2$km in my tests. This parameters is set so trips are geographically far apart, and so have a low likelihood of being shareable are eliminated. However, there will be cases where possible shared trips where the trips align but have far origin and destination. A more sophisticated geographic pruning method could be used which prunes trips outside a trip corridor which could reduce this over pruning and further work would be to conduct sensitivity analysis for this parameter.

Furthermore, the shareability graph is partitioned which does mean a suboptimal matching may be produced but community partitioning mitigates this.

> **Question 2**
>
> Implement your proposed method and evaluate Manhattan's overall efficiency using yellow taxi data from the first full week (Monday-Sunday) in June 2013. Discuss how your method would scale with more data; in other words, discuss the complexity of your implementation.

The algorithm was run for the Manhatten data for the first full week (Monday-Sunday) in June 2013, using the parameter values in Table 1. The choice of these values was fairly conservative. The time delay is fairly conservative at 6 minutes, but if we assume there is also some pickup and

Parameter	Value
Δ	360 s
C	6 persons

Table 1: Parameter values

dropoff delays which are unaccounted for this could be more around 15 minutes which is closer to most peoples delay threshold. VMT savings could be increased by increasing these parameters but this analysis has not been done here.

Results

- **Total trips processed:** 2,793,023
- **Total trips matched:** 2,484,582
- **Average match rate:** 88.96%
- **Absolute VMT savings:** 2,777,337 km
- **VMT savings:** 23.76%
- **Average Delay:** 35.8 sec

Pairwise aggregation at city scale achieves substantial efficiency: 23.76% VMT savings with strict delay ($\Delta = 6$ min) and capacity ($C = 6$) constraints.

Scalability, Partitioned Matching, and Complexity

Let n be trips per day, u unique trip nodes, p candidate pairs after pruning, and e edges in the shareability graph.

- **Mapping to nodes:** Vectorized nearest-node on G; near-linear in n (practically $O(n)$ with spatial index).
- **Shortest-path:** For each unique origin node, run Dijkstra on G; total $O(u\,(|E|+|V|\log|V|))$ in the worst case, but restricted nodes and persistence mean this is only calculated once. .
- **Vectorized pruning:** Batch temporal/capacity/geographic checks; worst-case $O(n^2)$ but with pruning yielding $p \ll n^2$ in practice.
- **Feasibility checks:** Sequential over p candidates with $O(1)$ lookups from the precomputed cache; $O(p)$ time.
- **Partitioning share ability graph.** We first decompose the shareability graph into connected components in $O(|V| + |E|)$ time; components with more than a target size T are further split by community detection (Louvain with a tuned resolution), both of which run in near-linear time per pass $O(|E|)$. Recursion ensures partitions satisfy $|V_k| \leq T$. This partitioning step is therefore near-linear overall.

- **Maximum-weight matching** For each partition $G_k = (V_k, E_k)$ we (i) build the rustworkx graph in $O(|V_k| + |E_k|)$ and (ii) run maximum-weight matching via a blossom-style solver (rustworkx `max_weight_matching`), which is $O(|V_k|^3)$ in the worst case. This improves the unpartitioned $O(n^3)$ bound when $T < n$.

Overall, the biggest term is $O(n^2)$ when building the shareability graph. Looser geographic thresholds or peak windows inflate p toward $O(n^2)$. With tight thresholds and effective partitioning, p is closer to linear in n, so it runs closer to $O(n)$.

In practical terms this could then be extended to cover longer time durations or a bigger area. However, if the frequency of trips were to be increased, in the same area and time window, then this would scale $O(n^2)$.

> **Question 3**
>
> Based on your implementation in the previous question, use visualizations to show how efficiency varies with time and location. Discuss any potential business implications based on your findings.

Figure 1: Hourly patterns by day of week (June 3–9, 2013). Each line is a day; panels show demand (top), shareability, match rate, system-wide VMT savings, and average delay.

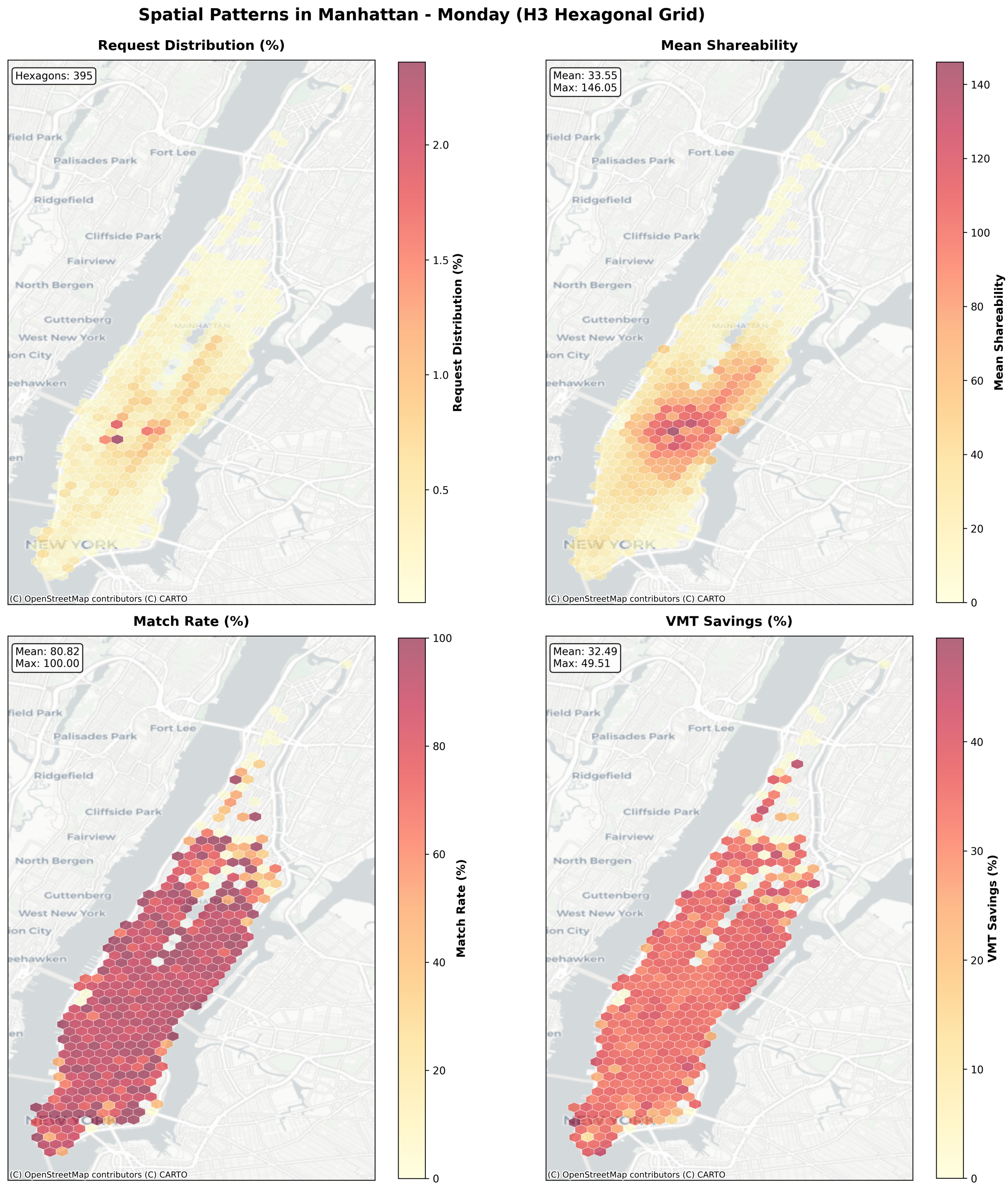

Figure 2: Spatial patterns by trip origin for 7am - 10am Monday 3rd June; panels show demand (top), shareability, match rate, VMT savings.

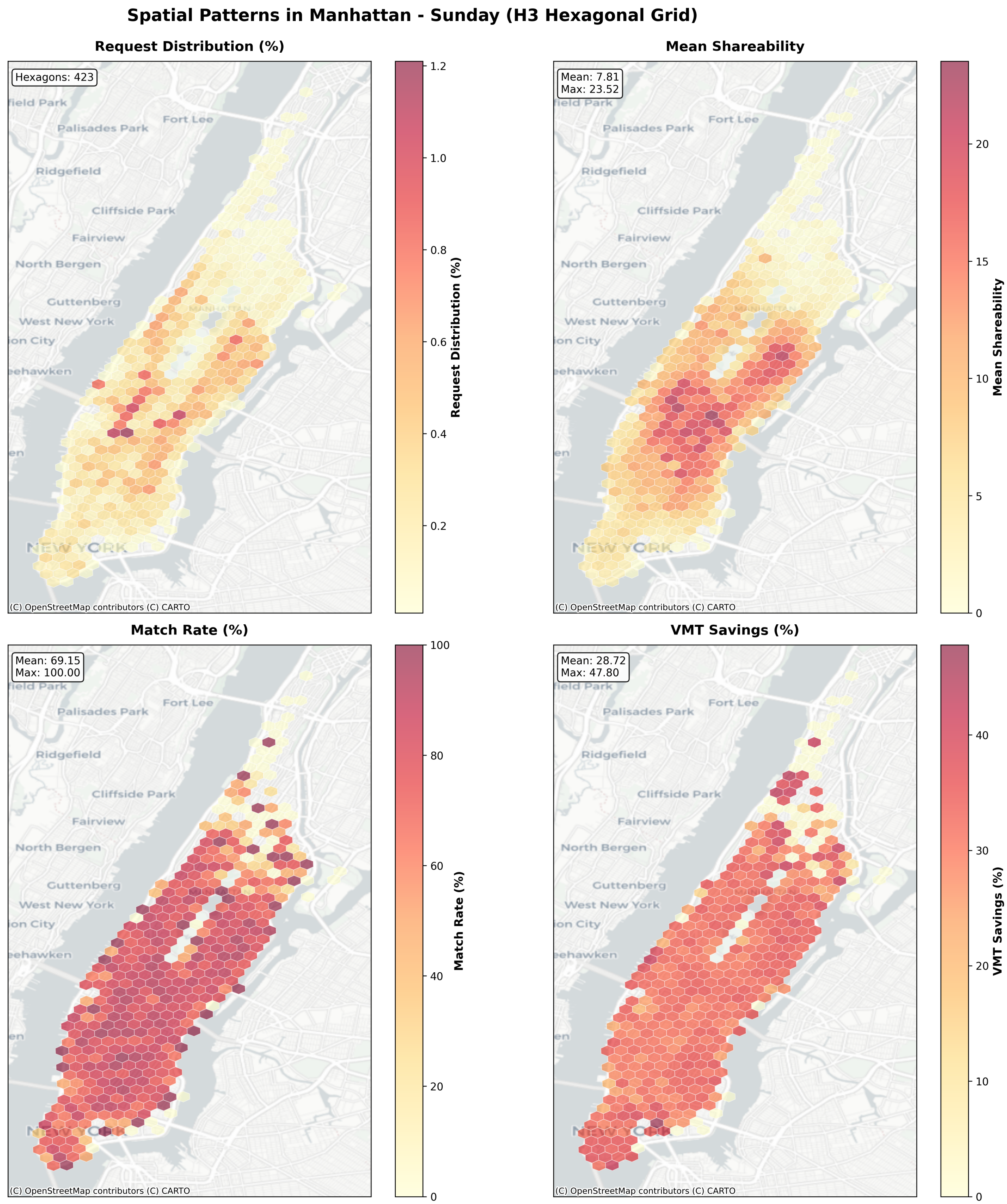

Figure 3: Spatial patterns by trip origin for 7am - 10am Sunday 9th June; panels show demand (top), shareability, match rate, VMT savings.

Temporal and Spatial findings

- Weekdays show the *highest* match rates (∼66%) and VMT savings (∼24%), which aligns with strong commuting regularity.
- Weekends show *lower* match rates and savings, reflecting more dispersed, leisure-oriented travel.
- Delays remain low across the week (median ≈ 0.66 min).
- Weekday morning and evening sees the highest trip requests. We see the highest shareability in the mornings which aligns with most commuters traveling to work at a consistent time in the morning but having more varied leaving times in the evening.

Interpretation. Time-of-day efficiency closely follows commuting structure. When trips are numerous and similar (morning and evening peaks), both *shareability* and *match rate* are high, sustaining *system-wide VMT savings* at elevated levels throughout the business day. Overnight, efficiency potential per-trip can still be meaningful on a percentage basis but contributes less in absolute terms due to very low volume. The geographic location shows the more strongly commuter concentration in the center of Manhattan giving a higher shareability degree where for Sunday it is more dispersed which aligns with peoples more varied leisure travel.

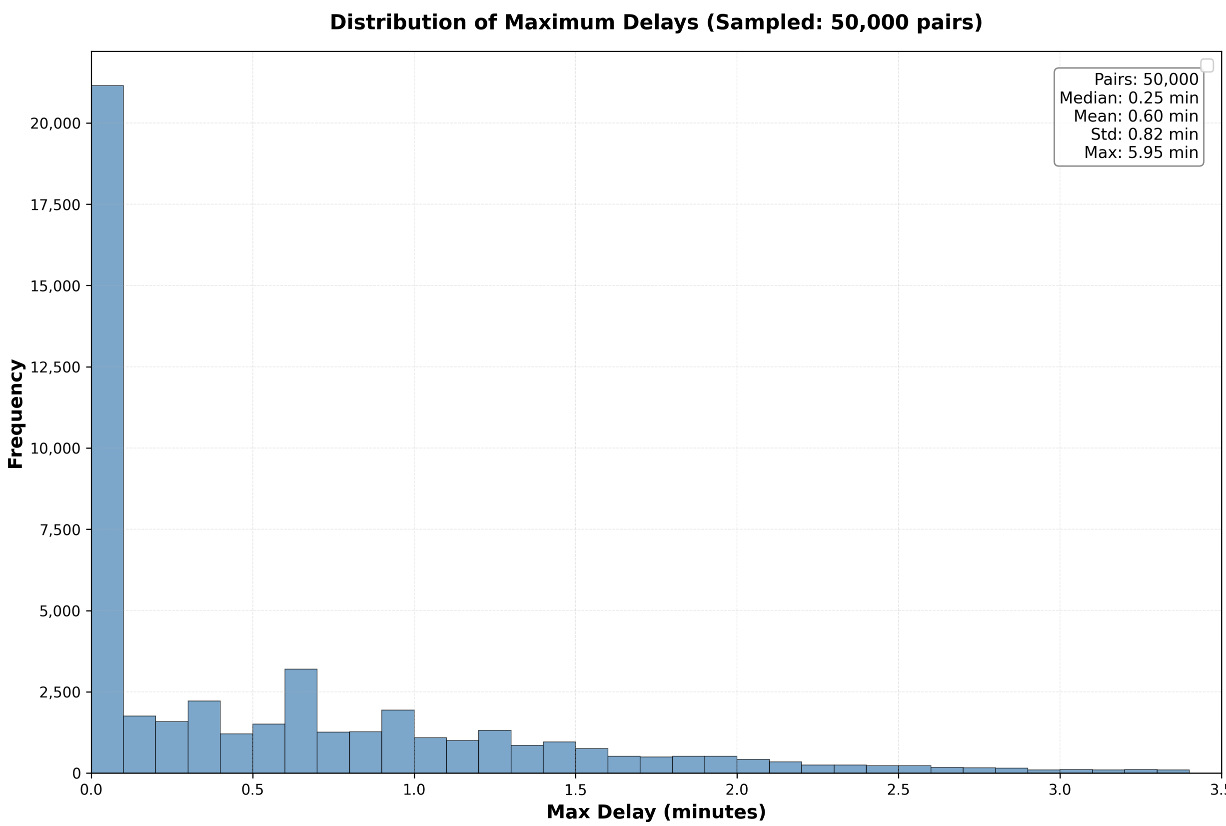

Figure 4: Distribution of trip delays across the week.

Low delays Firstly, the low mean delay is unrealistically low for a microtransit service when you factor in actual operational conditions as mentioned previously. Secondly, Figure 4 shows that most delays are very small, implying that the majority of matched trips are very similar, which could be expected for very regularized journeys, however, this could be a result of some possible routes, which are less similar but still within QoS constraints being pruned by the geographic filter, which should be adjusted in future work.

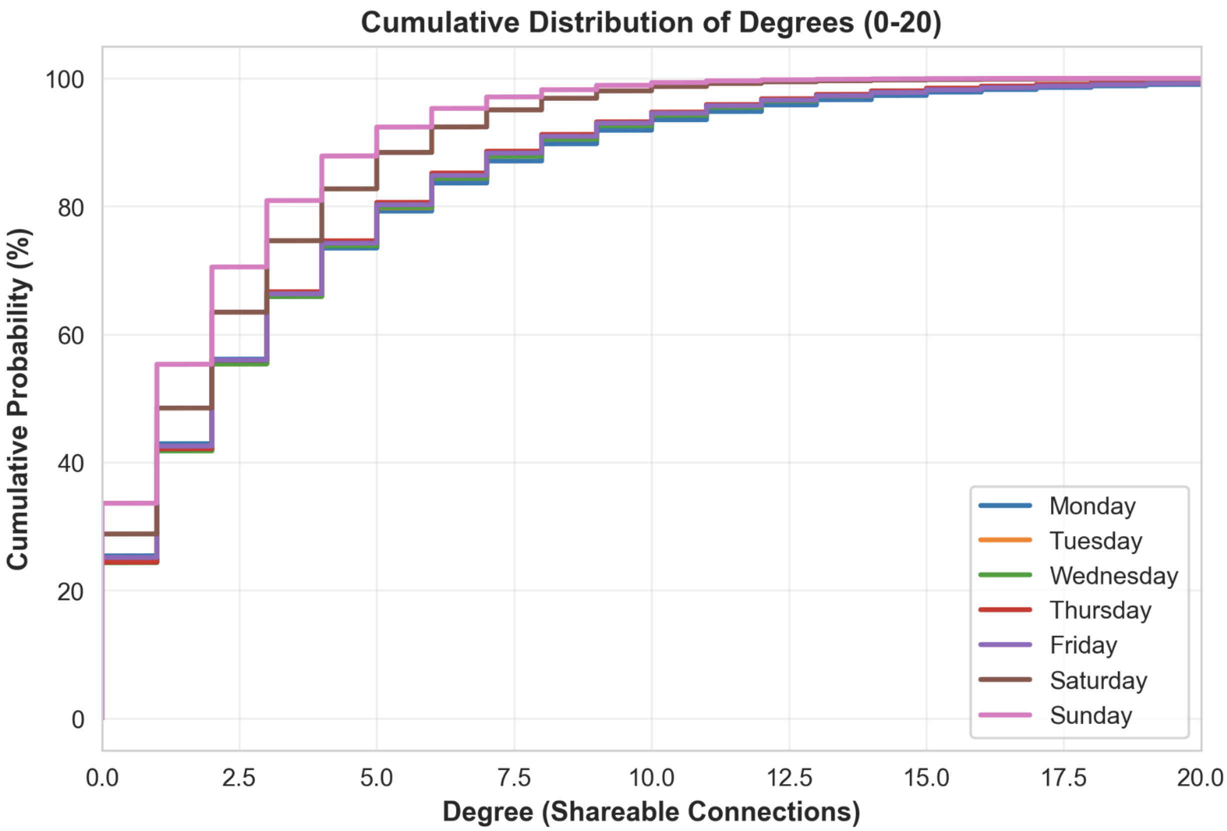

Figure 5: Cumulative distribution showing trips with more than X shareable trips.

Consistent Matching Rate and VMT Savings We can see how there is a disconnect between the shareability graph connectivity and the matching and VMT Saving. This is a result that despite being less connected Figure 5 shows that the percentage of trips with 0 or 1 shareable connections (which is where we expect most the non-matched requests to come from) for the weekends is with 25% of the weekday mean. It appears that there is a threshold when having a more connected shareability graph doesn't produce significantly greater VMT Savings using this algorithm. However, for a practical online application where demand is stochastic, having a high degree of shareability would be significant for making a more robust algorithm as there could be alternative allocations.

Business implications. Spatially, high-degree corridors, like major roads, deliver the most shareability and matches, whereas peripheral/late-night demand is more dispersed and less efficient. Operationally, concentrate fleet on peak windows, stage vehicles at corridor micro-hubs, use time/zone-based pricing to nudge pooling where match odds are highest, reposition between peaks.

0.1 Further Work

As mentioned previously, investigation into the geographic pruning to ensure shareable trips are not being excluded. Analysis could be done to examine the sensitivity to the QoS parameters. The algorithm could be extended to combine > 2 requests; a simple approach could by adding an insertion stage to test whether unmatched trips could be added to routes. This analysis does not include vehicle fleet size and even a simplified simulation assigning matched pairs to a fixed fleet using nearest-available-vehicle dispatching would reveal what VMT savings can be achieved with different fleet sizes.

References

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467.

Cramer, J. and Krueger, A. B. (2016). Disruptive change in the taxi business: The case of uber. *American Economic Review*, 106(5):177–82.

Paepe, W., Lenstra, J., Sgall, J., Sitters, R., and Stougie, L. (2004). Computer-aided complexity classification of dial-a-ride problems. *INFORMS Journal on Computing*, 16:120–132.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., and Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences*, 111(37):13290–13294.