Ride-hailing Service Optimization: A Multi-Agent Attention-Based Approach

Gyuseok Lee, Hyunkuk Lim, Sumin Lim

Table of Contents

1. Introduction

2. Previous Works

3. Our Approach

4. Experiments & Results

5. Conclusion

Introduction

Ride-Hailing Service Research

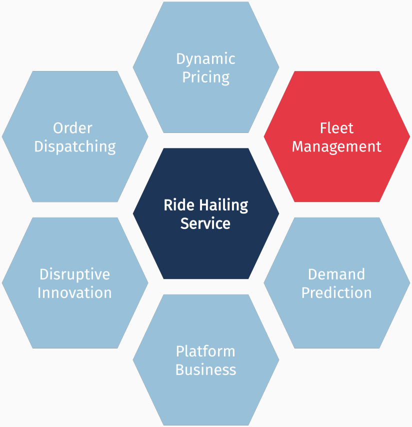

Figure 1: Research on Ride-Hailing Service

Fleet Management

- Finding efficient means of dispatching vehicles to minimize mismatching rate in the future

Previous works

- Focused on solving overall difference between demand and supply of ride requests
- Strong assumptions are applied to reduce computations
  - Simplification of driver agents
  - Restriction of state space through limiting the distance a vehicle can be repositioned
- Ride-hailing platform that recommends the most suitable orders to the drivers cannot work well based on one-to-one matching dispatching model (Chen et al., 2022)

Previous Works

Types of Agent

Driver-based Approach

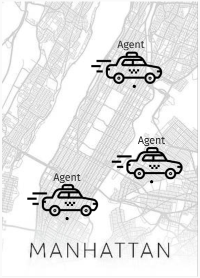

Figure 2: Agent

Grid-based Approach

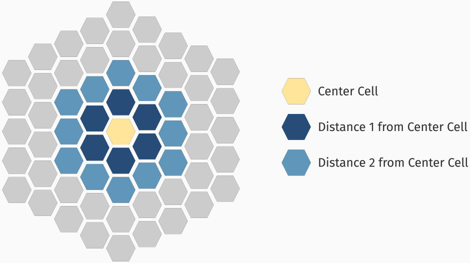

Figure 3: Grid

Efficient Large-Scale Fleet Management (Lin et al., 2018)

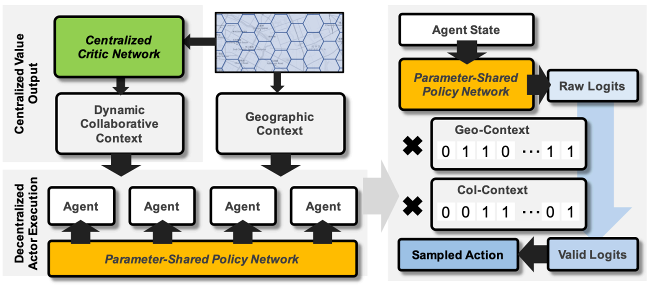

Figure 4: Overall Architecture

Weakness: Drivers are considered agents, but all drivers in a single grid are assumed to be homogeneous

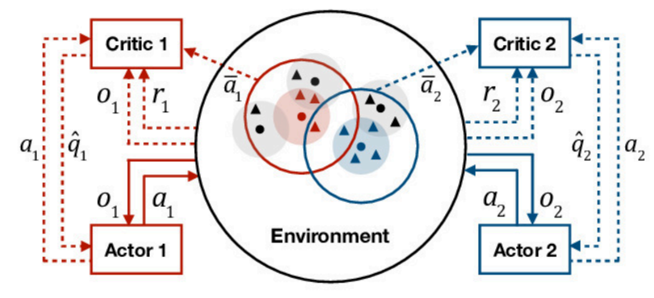

Figure 5: Overall Architecture

Weakness: Driver agents are also homogeneous, and local interactions through a mean field leads each action to have little impact on the outcome, grouping the agents according to their respective fields

CoRide (Jin et al., 2019)

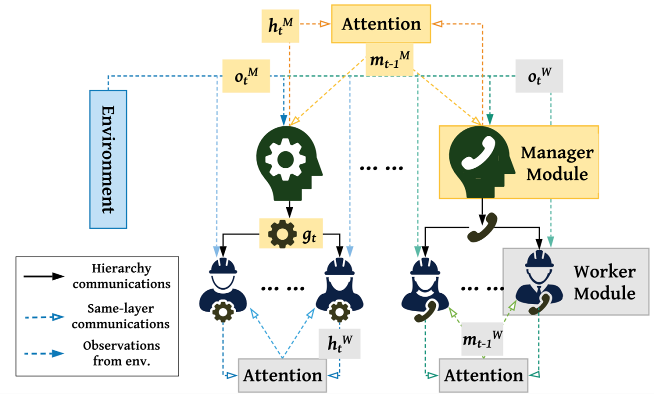

Figure 6: Overall Architecture of CoRide

Weakness: The maximum distance that a vehicle can be repositioned at maximum is two grids

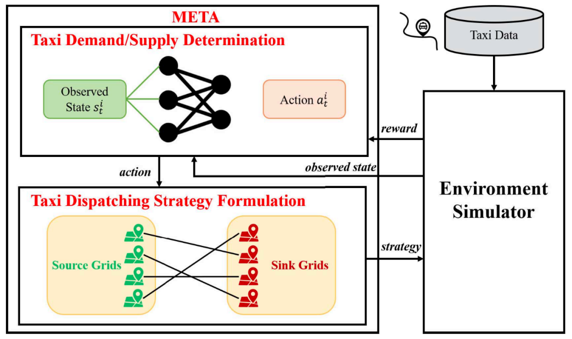

Figure 7: Overall Architecture of META

Weakness: No limitation in the distance that a vehicle can move, but the reposition is restricted to one-to-one match between source and sink grids

Our Goal

Based on the Liu et al. (2021)'s approach,

- Replace one-to-one matching to many-to-many matching
  - Ride-hailing platform that recommends the most suitable orders to the drivers cannot work well based on one-to-one matching dispatching model (Chen et al., 2022)
- Methods for many-to-many matching
  - Regard repositioning problem as minimum-cost network flow problem
- Compare one-to-one matching and many-to-many matching
  - Employ Kuhn-Munkres algorithm for one-to-one matching

Our Approach

Preliminaries I

- **Agent**: A city grid, $i \in [1, \ldots, N]$
- **State**: $s_i^t = Num_d^i - Num_s^i$
  - Num_d^i: Number of taxi demands in an agent i
  - Num_s^i: Number of taxi supplies in an agent i
- **Action**: Number of taxis that needs to dispatch to nearby grids
  - $|a_t^i \times Num_s^i|$ taxis need to be repositioned, $a_t^i \in [-1, 1]$
  - **Source grid**: An agent i if $a_t^i > \xi$ for any given threshold ξ
  - **Sink grid**: An agent i if $a_t^i < -\xi$
  - Do nothing if $|a_t^i| \leq \xi$

Preliminaries II

- **Strategy**: A set of repositioning pairs
    - A pair of a source grid and a sink grid
    - Determined by their corresponding actions
- **Reward**: Revenue that an agent i obtains at the time t

$$r_t^i = 1 - \frac{|Num_d^i - Num_s^i|}{\max(Num_d^i, Num_s^i)}$$

- Reflects the equilibrium degree of the agent i
- An agent aims to maximize the accumulated reward with the discount of γ, denoted as $\sum_{k=0}^{\infty} \gamma^k r_{t+k}^i$
- **Heterogeneous Taxis**
    - Taxis in the same grid can have two different relocation destinations - stay still or move to other grids

Assignment Problem Find a perfect match on a weighted complete bipartite graph $G = (S, T, E)$ with a minimum total cost

- S: Source vertices
- T: Target vertices
- E: Edges from source to target
- Each edge has a nonnegative cost $c(i, j)$

The Kuhn-Munkres Algorithm (also known as the Hungarian method).

Start with an arbitrary feasible vertex labeling l, determine G_l, and choose an arbitrary matching M in G_l.

1. If M is complete for G, then M is optimal. Stop. Otherwise, there is some unmatched $x \in X$. Set $S = \{x\}$ and $T = \emptyset$.

2. If $J_{G_l}(S) \neq T$, go to step 3. Otherwise, $J_{G_l}(S) = T$. Find

$$\alpha_l = \min_{x \in S, y \in T^c} \{l(x) + l(y) - w(xy)\}$$

where T^c denotes the complement of T in Y, and construct a new labeling l' by

$$l'(v) = \begin{cases} l(v) - \alpha_l & \text{for } v \in S \\ l(v) + \alpha_l & \text{for } v \in T \\ l(v) & \text{otherwise} \end{cases}$$

Note that $\alpha_l > 0$ and $J_{G_{l'}}(S) \neq T$. Replace l by l' and G_l by $G_{l'}$.

3. Choose a vertex y in $J_{G_l}(S)$, not in T. If y is matched in M, say with $z \in X$, replace S by $S \cup \{z\}$ and T by $T \cup \{y\}$, and go to step 2. Otherwise, there will be an M-alternating path from x to y, and we may use this path to find a larger matching M' in G_l. Replace M by M' and go to step 1.

Figure 8: KM Algorithm

Minimum Cost Network Flow Problem (NF) I

- $G = (V, E)$: A directed graph
- $s \in V, t \in V$: A source vertex
- $t \in V$: A sink vertex
- $(u, v) \in E$: An edge
- $c(u, v) > 0$: Each edge's capacity
- $f(u, v)$: Flow from u to v
- $a(u, v)$: Cost from u to v

Minimum Cost Network Flow Problem (NF) II

Minimize the total cost of the flow over all edges

$$\sum_{(u,v)\in E} a(u,v) \cdot f(u,v)$$

such that

- $f(u,v) \leq c(u,v)$
- $f(u,v) = -f(v,u)$
- $\sum_{w\in V} f(u,w) = 0$ for all $u \neq s,t$
- $\sum_{w\in V} f(s,w) = d$ and $\sum_{w\in V} f(w,t) = d$

Our Approach I

- Divide grids into source and sink
  - Source: Supply exceeds demand (have enough taxis)
  - Sink: Demand exceeds supply (don't have taxis)
- Each sink grid sends requests to multiple source grids

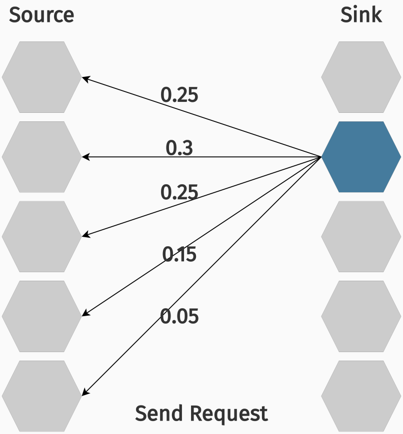

Figure 9: Many-to-many matching with Attention

Our Approach II

$$\text{score}(\mathbf{e}_{source}, \mathbf{e}_{sink}) = \begin{cases} \mathbf{e}_{source}^T \mathbf{e}_{sink} & \text{Dot-product} \\ \text{attention}(\mathbf{e}_{source}, \mathbf{e}_{sink}) & \text{QKV Attention} \end{cases}$$

, where

- $\mathbf{e}_{source}$: Embedding for sink grids
- $\mathbf{e}_{sink}$: Embedding for source grids

$$\text{Ratio} = \text{softmax}(\text{score}(\mathbf{e}_{source}, \mathbf{e}_{sink}))$$

Based on the Ratio, the sink grid requests excess demands proportionally to source grids

Experiments & Results

1. Algorithms
   - One-to-one matching
     - Kuhn–Munkres Algorithm
   - Many-to-many matching
     - Network flow
     - Attention
2. Maximum Distance: 2, 5, 10
3. Low resolution

Results I

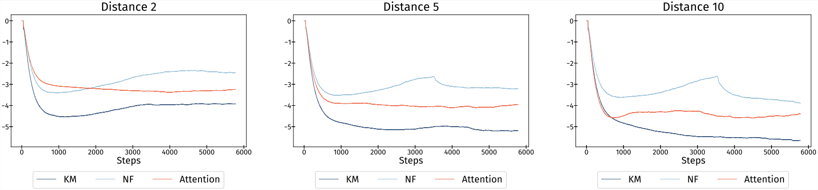

Figure 10: Actor Loss

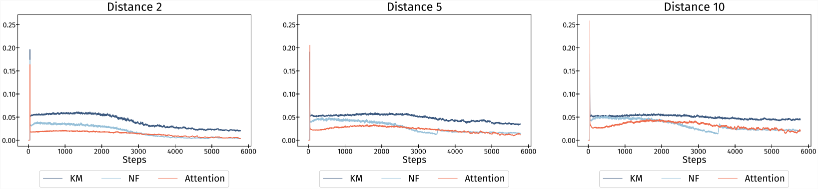

Figure 11: Critic Loss

Results II

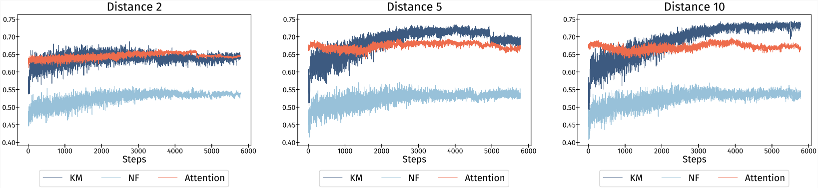

Figure 12: Matching Rate

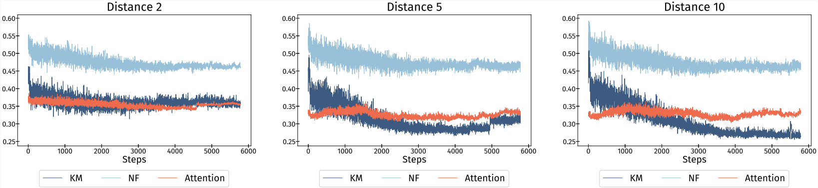

Figure 13: Mismatching Rate

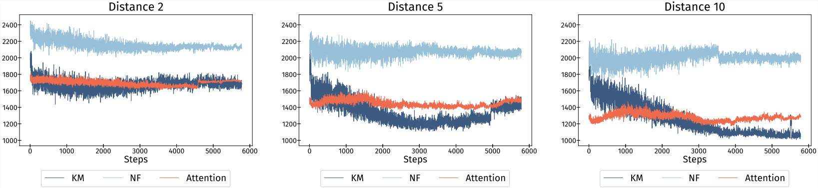

Figure 14: Average Unbalanced Supply and Demand Quantity

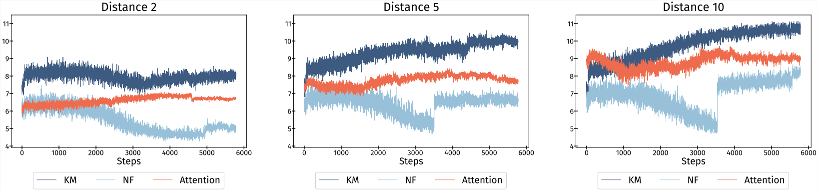

Figure 15: Average Reward

Experiment 2 - Settings

1. Algorithms
   - One-to-one matching
     - Kuhn–Munkres Algorithm
   - Many-to-many matching
     - Network Flow
     - Attention
2. Maximum Distance: 2, 5, 10
3. High resolution

Results I

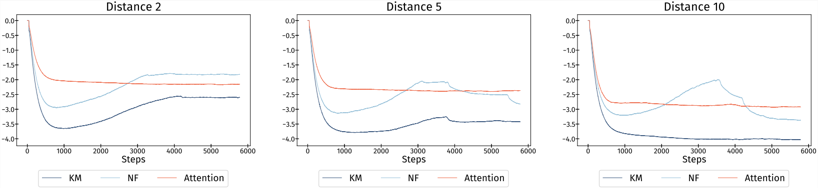

Figure 16: Actor Loss

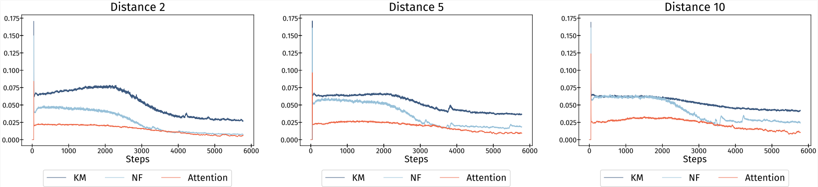

Figure 17: Critic Loss

Results II

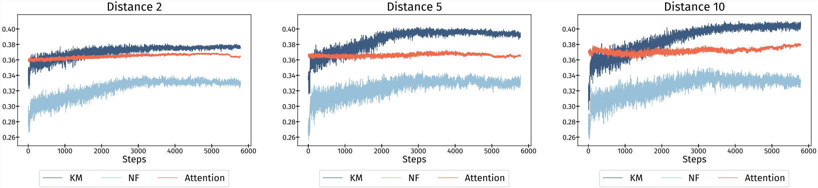

Figure 18: Matching Rate

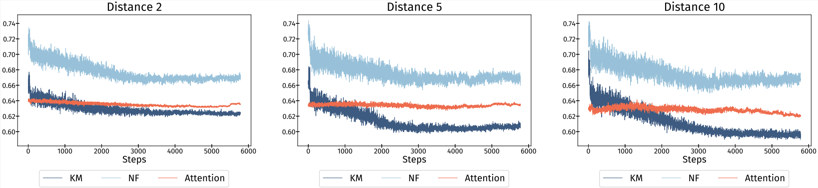

Figure 19: Mismatching Rate

Results III

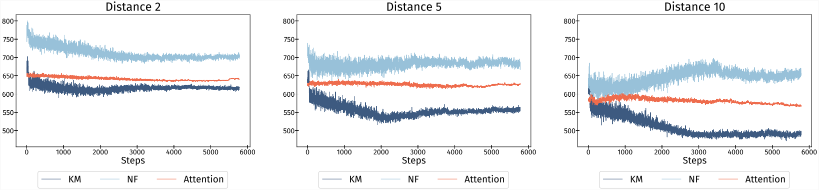

Figure 20: Average Unbalanced Supply and Demand Quantity

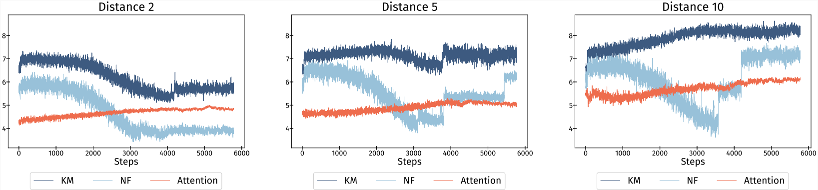

Figure 21: Average Reward

Conclusion

Limitations

1. Dispatching is only based on the demand needs of sink grids
    - Not taking into account the excess supply of the source grid, which leads to discrepancy in available vehicles sent to each sink grid
2. Reward assumes that dispatched drivers belong to the target sink grid, ignoring the time that vehicles need to transition from a source grid to a sink grid

Future Work

- Dispatch based on combinations of attention scores of source grids and sink grids
- Dispatching requests per source grids with respect to the number of demands for each sink grids
- Develop new reward functions for optimizing drivers’ profit based off of different possible repositioning strategies

Thank you for listening!

Any Questions?

References

Chen, H., Z. Li, and Y. Yao (2022). Multi-agent reinforcement learning for fleet management: a survey. In *2nd International Conference on Artificial Intelligence, Automation, and High-Performance Computing (AIAHPC 2022)*, Volume 12348, pp. 611–624. SPIE.

Jin, J., M. Zhou, W. Zhang, M. Li, Z. Guo, Z. Qin, Y. Jiao, X. Tang, C. Wang, J. Wang, et al. (2019). Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1983–1992.

Li, M., Z. Qin, Y. Jiao, Y. Yang, J. Wang, C. Wang, G. Wu, and J. Ye (2019). Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The world wide web conference*, pp. 983–994.

Lin, K., R. Zhao, Z. Xu, and J. Zhou (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1774–1783.

Liu, C., C.-X. Chen, and C. Chen (2021). Meta: A city-wide taxi repositioning framework based on multi-agent reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*.