Multi-Agent Reinforcement Learning for Dynamic Mobility Resource Allocation with Hierarchical Adaptive Grouping

Farshid Nooshi and Suining He
Ubiquitous & Urban Computing Lab
University of Connecticut
Storrs, CT, USA
{farshid.nooshi,suining.he}@uconn.edu

Abstract

Allocating mobility resources (e.g., shared bikes/e-scooters, ride-sharing vehicles) is crucial for rebalancing the mobility demand and supply in the urban environments. We propose in this work a novel multi-agent reinforcement learning named Hierarchical Adaptive Grouping-based Parameter Sharing (HAG-PS) for dynamic mobility resource allocation. HAG-PS aims to address two important research challenges regarding multi-agent reinforcement learning for mobility resource allocation: (1) how to dynamically and adaptively share the mobility resource allocation policy (i.e., how to distribute mobility resources) across agents (i.e., representing the regional coordinators of mobility resources); and (2) how to achieve memory-efficient parameter sharing in an urban-scale setting.

To address the above challenges, we have provided following novel designs within HAG-PS. To enable dynamic and adaptive parameter sharing, we have designed a hierarchical approach that consists of global and local information of the mobility resource states (e.g., distribution of mobility resources). We have developed an adaptive agent grouping approach in order to split or merge the groups of agents based on their relative closeness of encoded trajectories (i.e., states, actions, and rewards). We have designed a learnable identity (ID) embeddings to enable agent specialization beyond simple parameter copy. We have performed extensive experimental studies based on real-world NYC bike sharing data (a total of more than 1.2 million trips), and demonstrated the superior performance (e.g., improved bike availability) of HAG-PS compared with other baseline approaches.

CCS Concepts

• **Computing methodologies** → **Reinforcement learning**; • **Information systems** → *Spatial-temporal systems.*

Keywords

Multi-agent reinforcement learning, dynamic parameter sharing, mobility resource rebalancing, hierarchical grouping

1 Introduction

Dynamic allocation of urban mobility resources, such as the shared bikes [10], e-scooters [3], and ride-sharing vehicles [1, 8], is crucial for enhancing the operational efficiency of urban mobility systems and satisfying the mobility needs of various communities. Due to the complex city environments and varying mobility needs, how to adaptively rebalance the demands and supplies is crucial for the success of allocation. Among various approaches explored to support city-scale rebalancing, multi-agent reinforcement learning (MARL) has been explored due to its adaptivity and scalability. By considering coordinators (e.g., ride sharing drivers, bike sharing relocators) as agents, MARL can serve as the resource distribution engine, observing the mobility resources and environment (states), and dynamically allocate the mobility resources (actions) through strategically (policy) coordinating these agents' behaviors.

Despite the prior results, two major challenges remain before the MARL can be deployed for mobility resource allocation and demand-supply rebalancing. *First*, how can we dynamically and adaptively share the mobility resource allocation policy (e.g., when and where the available mobility resources should be re-allocated) across different agents that are coordinating? One may consider grouping the agents and respectively instantiating a set of policy networks for each agent group [18]. Such a grouping method, however, may not provide a principled mechanism on the group number and their sizes. Therefore, sub-optimal performance may be achieved given the dynamic urban mobility demands and supplies. *Second*, how can we achieve scalable and memory-efficient parameter sharing in a city-wide setting? Learning the mobility allocation policy for every agent is not feasible. On the other hand, a shared mobility resource allocation policy may not necessarily capture heterogeneous roles or location-specific specialization of the agents. The performance of MARL may hence deteriorate given the complex urban environment with varying mobility demands and supplies [21].

To overcome the above-mentioned challenges, we propose in this work a novel multi-agent reinforcement learning named HAG-PS, Hierarchical Adaptive Grouping-based Parameter Sharing for dynamic and adaptive mobility resource allocation. Toward development of HAG-PS, we have made the following contributions:

- *Hierarchical Parameter Sharing with Scalable and Memory-Efficient Designs*: We have designed and developed a hierarchical parameter sharing mechanism for MARL. Our mechanism consists of global and local information of the mobility resource states (e.g., distribution of mobility resources across different regions). This way, our HAG-PS can enable dynamic and adaptive parameter sharing in a city-wide setting.
- *Adaptive Grouping of Coordinating Agents*: We have developed an adaptive parameter budget-capped agent grouping approach in order to split or merge the groups of agents based on their relative closeness of encoded trajectories (i.e., states, actions, and rewards). We have designed a learnable identity (ID) embeddings to enable agent specialization beyond simple parameter copy. Through these measures, we

enhance the model adaptivity in dynamic mobility resource allocation.
- *Extensive Emulation Studies based on Bike Sharing Mobility Data*: We have performed extensive experimental studies based on real-world NYC bike sharing data (a total of 1,232,838 trips), and demonstrated the superior performance (e.g., improved bike availability and rebalanced bikes) of HAG-PS compared with other baseline approaches.

The rest of the paper is organized as follows. We first present the related work in Section 2. Then, we discuss the concepts, problem formulation, and core designs in Section 3. After that, we demonstrate the preliminary results in Section 4, and conclude with future studies in Section 5.

2 Related Work

We briefly review the related work in the following two categories.

• **Parameter Sharing for MARL.** Various parameter sharing methods have been explored in order to improve the memory requirements and learning efficiency of MARL [5–7]. In order to enhance the diversity of agent behaviors [21] of MARL instead of homogeneous roles, the agents can be grouped selectively [12, 19] based on similar trajectories. For instance, SePS [2] performs a one-shot clustering and reuses weights within each cluster. However, a major limitation of these approaches lies in that the role assignment of agents remain largely static — that is, once the role (group) is decided, an agent cannot be re-assigned. Therefore, these approaches may not adapt to the evolving urban mobility environment. DyPS [18] periodically re-clusters agents based on latent trajectories (e.g., states, actions, rewards), and support the urban resource allocation. However, as each group of agents holds its own policy network, it remains difficult to further enhance the scalability of the approach given large number of agents and groups in practice.

• **Urban Mobility Resource Allocation.** Reinforcement learning [9, 20], including MARL, has been explored for bike sharing resource allocation [14, 16], traffic light control [4], and ride-sharing [13]. For instance, i-Rebalance [13] studied shared-policy reinforcement learning for city-wide repositioning of idle ride-sharing vehicles. However, these approaches often rely on heuristic clustering or operate with one shared policy, which limits their adaptability in the complex urban environments.

• **Differences from Prior Arts.** Different from the above studies, HAG-PS advances from the following aspects. First, we have enabled a low-dimensional identify (ID) embedding as well as the model weights for every agent. This way, we can enhance the invidual learnability of the agents [17]. Second, HAG-PS provides a dynamic adjustment of agent role (group), enabling city-wide mobility resource allocation.

3 Concepts, Problem Formulation, & Core Designs

3.1 Concepts & Problem Formulation

• **Spatial & Temporal Discretization.** Following the prior studies, we discretize the service area into a total of K rectangular regions that are indexed by $\mathcal{K}=\{1,\ldots,K\}$. A set of N agents serving as the mobility resource re-allocators (e.g., a fleet of trucks or coordinators for bike rebalancing) is denoted by $\mathcal{A}=\{1,\ldots,N\}$ where each agent i serve a region at each time interval t. We discretize the time horizon into time intervals (e.g., days in our current studies).

• **States & Actions.** We design the mobility resource states for HAG-PS to capture and learn. Specifically, for each time interval t, we have a global state $\mathbf{s}_t$ which consists of mobility features (i.e., time of a day, day of a week, distributions of available mobility resources, historical pick-up statistics) and urban environment features (harvested from OpenStreetMap). In this prototype study, we encode the time of a day (h_t in hour), day of a week d_t (integers from 0 to 6) of an time interval t into a vector of $[\sin\frac{2\pi h_t}{24}, \cos\frac{2\pi h_t}{24}, \sin\frac{2\pi d_t}{7}, \cos\frac{2\pi d_t}{7}]$. We take in the availability of mobility resources (e.g., number of available bike inventory) per region as $[b_t^1, b_t^2, \ldots, b_t^K]$. For each region k at the time interval t, we find the means μ_t^k and standard deviations σ_t^k of aggregate pick-ups (from all stations in a region) in the most recent H time intervals, and therefore we have for all regions the statistics vector $[\mu_t^1, \sigma_t^1, \mu_t^2, \sigma_t^2, \ldots, \mu_t^K, \sigma_t^K]$. For each region k, we encode the numbers of roads, bike lanes, and POIs within the region into a three-dimension vector $[\mathrm{rd}^k, \mathrm{bd}^k, \mathrm{pd}^k]$ [15].

For each agent i at time interval t, we find the sizes of mobility resources (e.g., bikes) that will relocate from the current region to one of the four adjacent regions (to the north, south, east, and west) as $\mathbf{a}_t^i = [a_t^{(\mathrm{North},i)}, a_t^{(\mathrm{South},i)}, a_t^{(\mathrm{East},i)}, a_t^{(\mathrm{West},i)}]$. The value in $\mathbf{a}_t^i$, if negative, represents the relocation from an adjacent region.

Let d_t^k and o_t^k respectively be the numbers of pick-up requests and drop-offs at the region $k \in \mathcal{K}$ at the time interval t. Given the states and actions, HAG-PS finds and updates the availability of mobility resources for each region as

$$b_{t+1}^k = \max\left\{ b_t^k + \sum_{j:\mathrm{loc}(j)\in\mathcal{N}(k)} a_t^{(*,j)} - d_t^k + o_t^k, 0 \right\}, \tag{3.1}$$

where $\mathcal{N}(k)$ represents the adjacent regions of a region k.

• **Reward Function.** Given a coordinated relocation decision $\mathbf{A}_t = \{\mathbf{a}_t^1, \mathbf{a}_t^2, \ldots, \mathbf{a}_t^N\}$ from N agents, the availability of mobility resources (inventory) in a region k (before serving the actual pick-up requests) is

$$\tilde{b}_t^k = b_t^k + \sum_{j:\mathrm{loc}(j)\in\mathcal{N}(i)} a_t^{(*,j)}, \tag{3.2}$$

where $a_t^{(*,j)}$ is the signed net inflow contributed by agent i. The fulfilled demand is then given by

$$S_t^k = \min\{d_t^k,\ \tilde{b}_t^k\}, \tag{3.3}$$

while the unfulfilled demand is calculated by

$$U_t^k = d_t^k - S_t^k = \max\{d_t^k - \tilde{b}_t^k, 0\}. \tag{3.4}$$

We define the reward function for each agent i at time interval k as

$$r_t^i = \lambda\left(1 - \frac{U_t^i}{d_t^i+\epsilon}\right) - \alpha\,\frac{U_t^i}{d_t^i+\epsilon} - \beta\,\frac{\|\mathbf{a}_t^i\|_1}{m}, \tag{3.5}$$

where the three components ($\lambda, \alpha, \beta > 0$) inside the reward function r_t^i respectively favor (i) high service ratio, (ii) low under-served demand, and (iii) low cost of relocation that is proportional to the total size of relocated mobility resources capped by maximum relocation load $m > 0$.

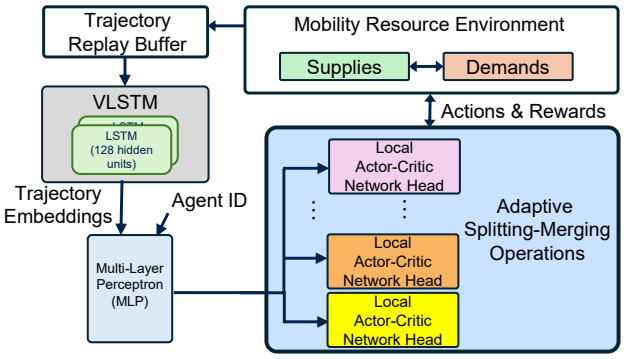

Figure 1: Illustration of overall architecture of HAG-PS.

• **Problem Formulation.** We formulate our mobility resource allocation problem as a finite–horizon multi-agent Markov game. For a horizon of T time intervals, and a discount factor $\gamma \in (0, 1]$, HAG-PS aims to find the MARL parameters $\boldsymbol{\theta}$ that maximize the objective function of

$$J(\boldsymbol{\theta}) = \mathbb{E}\Big[\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\gamma^t r_t^i\Big]. \tag{3.6}$$

3.2 Hierarchical Adaptive Grouping

We have designed the hierarchical adaptive grouping to dynamically assign the group (role in mobility resource allocation) for each agent. We have designed adaptive splitting and merging methods, such that the number of groups and the sizes of each group dynamically change as the learning evolves.

To better serve the urban-scale setting, the global group can serve as the macro coordination for a district or a county. Each global group indexed by $C_t^{(c)}$ at the interval t has a feature trunk network $T_{\boldsymbol{\theta}_c}$ shared across all the agents inside the group c. The resulting embeddings $\mathbf{h}_t^i = T_{\boldsymbol{\theta}_c}(\mathbf{s}_t)$. Within each group, we further partition multiple local groups, each of which (c, l) maintains a local actor–critic network head which has compact structure. Such local groups can serve underneath the global group as micro coordination for the neighborhoods or street levels. The local actor-critic network head of the agent i is formed by a multi-layer perceptron (MLP) with 128 units and maps the concatenated vector of $\mathbf{h}_t^i$ and identity embeddings $\mathbf{e}_i$ into the action $\mathbf{a}_t^i$ and reward r_t^i.

• **Dynamic & Adaptive Agent Grouping.** After every episode, each agent encodes its latest H-step trajectory

$$\boldsymbol{\tau}_{t-H:t}^{\,i} = \left\{\mathbf{s}_{t-H}^i, a_{t-H}^i, r_{t-H}^i, \ldots, \mathbf{s}_{t-1}^i, a_{t-1}^i, r_{t-1}^i\right\}, \tag{3.7}$$

via a Variational Long Short-term Memory (VLSTM) with 128 hidden units and obtain the embeddings, i.e.,

$$\mathbf{z}_t^i = \texttt{VLSTM}\left(\boldsymbol{\tau}_{t-H:t}^{\,i}\right). \tag{3.8}$$

For each local group l, let $|\mathcal{G}_t^{(l)}|$ be its group size, and we find the average embeddings

$$\mu_t^{(l)} = \frac{1}{|\mathcal{G}_t^{(l)}|}\sum_{j\in\mathcal{G}_t^{(l)}}\mathbf{z}_t^j, \tag{3.9}$$

and the average symmetrized KL divergence within the group,

$$D_t^{(l)} = \frac{1}{|\mathcal{G}_t^{(l)}|}\sum_{j\in\mathcal{G}_t^{(l)}}\tfrac{1}{2}\big(\mathrm{KL}(\mathbf{z}_t^j \,\|\, \mu_t^{(l)}) + \mathrm{KL}(\mu_t^{(l)} \,\|\, \mathbf{z}_t^j)\big). \tag{3.10}$$

Splitting and Merging Operations. The splitting and merging operations of the agents are given as follows. If $D_t^{(l)} > D_{\text{split}}$ and $|\mathcal{G}_t^{(l)}|$ is greater than the pre-defined group size $S_{\min}$, we bisect an agent group l with k-means (k=2) over $\{\mathbf{z}_t^i\}$. Both local groups inherit the parent's local actor-critic network heads, and immediately re-cluster their members into a total of $S_{\max}$ sub-groups. Two local groups inside the same global group are merged if $\mathrm{KL}(\mu_t^{(a)}||\mu_t^{(b)}) + \mathrm{KL}(\mu_t^{(b)}||\mu_t^{(a)})$ is less than a pre-defined merging threshold τ_{merge}. The local actor-critic network heads of the larger sub-group are kept, while the agents are re-assigned via k-means to the nearest of the $S_{\max}$ centroids.

Adaptive Regrouping Period. In order to enable adaptive regrouping period instead of a fixed one, we find a running exponential average $\bar{D}_t = \eta\,\bar{D}_{t-1} + (1-\eta)\frac{1}{|\mathcal{G}_t^{(l)}|}\sum_{l=1}^{|\mathcal{G}_t^{(l)}|} D_t^{(l)}$ to adjust the period before the next split–merge operations, i.e.,

$$\Delta_{t+1} = \max\Big(1,\ \big\lceil \Delta_0\, e^{-\zeta(\bar{D}_t - \delta)}\big\rceil\Big)\,. \tag{3.11}$$

Thus, when KL divergence within the local group has stabilized, regrouping becomes infrequent. As we can bound the maximum numbers of global group and local groups, we can restrict the model parameters and subsequent memory footprint of the feature trunk network and the local actor-critic network head. Our future extension will include detailed theoretical analysis over the performance.

4 Experimental Evaluation

We present our experimental settings, baselines, and experimental results as follows.

4.1 Experimental Settings

• **Dataset Preparation.** We leverage a total of 1,232,838 trips in January 2024 for our experimental studies. We have aggregated them into an origin–destination matrix over a total of $K = 106$ 1x1 km^2 rectangular regions covering the central Manhattan.

• **Comparison Baselines.** We compare our approach with the following baseline approaches.

- **No-Share**: which coordinates the mobility resources with fully independent proximal policy optimization (PPO).
- **Share-All**: which has one global actor–critic shared by all agents.
- **SePS** [2]: which performs offline k-means clustering for agent grouping.
- **DyPS** [18]: which performs dynamic grouping with group networks.
- **CDS** [12]: which performs diversity-regularized full parameter sharing.
- HAG-PS **w/o ID**: which removes the identity (ID) embeddings from HAG-PS.
- HAG-PS **w/o SM**: which performs no split–merge operation (i.e., fixed group sizes).
- HAG-PS **w/o HG**: which performs no hierarchical grouping.

- HAG-PS **w/o ARP**: which performs no adaptive regrouping period.

• **Implementations & Detailed Parameter Settings.** All models are implemented in PyTorch and trained on a single T4 GPU (16 GB RAM) on Google Colab. The PPO implementation follows the 37-detail checklist [11] (mini-batch SGD, value-loss clipping, etc.). We configure the model and environment settings to balance performance and training stability. The reward function employs weighting coefficients $\alpha = 5.0$, $\beta = 15.0$, and $\gamma = 3.0$ to respectively emphasize fulfillment of demand, penalize underservice, and discourage excessive relocations. Environment inputs include temporal encodings with 4 dimensions, spatial features with 6 dimensions, and one-step demand history represented by a single dimension. Each training epoch consists of 64 episodes (corresponding to 64 simulated months), and each episode spans a maximum of 31 decision steps to cover the days in January. We apply PPO with a discount factor $\gamma = 0.995$ and generalized advantage estimation (GAE) with $\lambda = 0.95$. The learning rates for policy and value networks are set to `3e-4` and `1e-3`, respectively to reflect a more conservative update for the policy to ensure stable learning while allowing faster adaptation of the value function.

To ensure robust evaluation, we reserve 20% of the training data for validation purposes. The split–merge controller is governed by five stable hyper-parameters. The regroup interval is initialized to $\Delta_0 = 8$ episodes and clipped to the range $\Delta \in [1, 64]$. After each episode the running KL divergence is updated with an exponential smoother ($\eta = 0.90$) and compared with the target drift level ($\delta = 0.02$); any excess shortens the next interval with sensitivity $\zeta = 3$. This configuration balances rapid reaction to behavioral shifts against the computational overhead of regrouping.

Evaluation Metrics. Following the prior practices in bike resource reallocation [6, 16], we evaluate overall performance via (1) fulfilled service ratio, i.e.,

$$\text{Avail}_T = 1 - \frac{\sum_{t=0}^{T-1}\sum_{k=1}^{K} U_t^k}{\sum_{t=0}^{T-1}\sum_{k=1}^{K} d_t^k}, \tag{4.1}$$

i.e. the fraction of all pickup requests that are successfully served over an episode; and (2) the total number of bikes that get rebalanced. Higher fulfilled service ratio and total bike rebalanced indicates better performance in mobility resource allocation.

4.2 Preliminary Experimental Results

Table 1 summarizes the fulfilled service ratio and the total number of bikes that get rebalanced. We can see that No-Share, which is the fully-independent PPO baseline, achieves only about 51 % fulfilled service ratio, demonstrating the drawback of over-parameterization without sharing the model parameters. Share-All reserves the model parameters but its performance falls below 44%. Compared with CDS, SePS, and DyPS, our achieves a higher fulfilled service ratio of 77.21% and more bikes rebalanced thanks to the gained learnability from its hierarchical adaptive grouping-based parameter sharing.

We also demonstrate the ablation studies within Table 1. We can see that removing the identity (ID) embeddings degrades about 0.3 percent in terms of fulfilled service ratio. This indicates that a small agent-specific vector can help disentangle similar mobility allocation policies. Disabling the splitting and merging operation or the hierarchical adaptive grouping leads to more performance degradation, and decreases fulfilled service ratios by about 2.1 percent and 4.0 percent, respectively. This underscores their importance for the mobility resource allocation. Fixing the regrouping period instead of an adaptive one leads to 1.1 percent performance drop in terms of fulfilled service ratio. Combining all these designs lead to overall superior performance of HAG-PS.

Table 1: Performance comparison of different schemes.

Method	Fulfilled Service Ratio (%)	Total Bikes Rebalanced
No-Share	51.18	357,864
Share-All	43.84	333,372
CDS	58.40	407,316
SePS	64.77	453,180
DyPS	69.09	462,696
HAG-PS w/o ID	76.91	471,900
HAG-PS w/o SM	75.10	470,028
HAG-PS w/o HG	73.23	468,936
HAG-PS w/o ARP	76.07	471,588
HAG-PS	**77.21**	**472,212**

5 Conclusion & Future Work

We study in this work a multi-agent reinforcement learning named hierarchical and adaptive grouping-based parameter sharing (HAG-PS) for dynamic mobility resource allocation. Using the NYC bike sharing as a case study, HAG-PS addresses two challenges regarding MARL for mobility resource allocation – that is, adaptive sharing of policy across agents, and memory-efficient parameter sharing in the urban-scale setting. we have designed a hierarchical approach that consists of global and local information of the mobility resource states. We have developed an adaptive budget-capped agent grouping approach to split or merge the groups of agents based on their relative closeness of encoded trajectories. Extensive experimental studies based on over 1.2 million bike sharing trips have validated the performance of HAG-PS in rebalancing the demand and supply in a metropolitan setting. Our future work will include: (i) expansion of experimental studies; (ii) introduction of multi-city data evaluations.

Acknowledgments

This project is supported, in part, by the National Science Foundation (NSF) under Grants 2239897 and 2303575, and the Connecticut Division of Emergency Management & Homeland Security (DEMHS) Hazard Mitigation Grant Program (HMGP). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Xiaohui Bei and Shengyu Zhang. 2018. Algorithms for trip-vehicle assignment in ride-sharing. In *Proc. AAAI*, Vol. 32.

[2] Filippos Christianos, Georgios Papoudakis, Muhammad A. Rahman, and Stefano V. Albrecht. 2021. Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing. In *Proceedings of the 38th International Conference on*

Machine Learning (ICML). 1989–1998.

[3] James C Chu, Hsin-Chia Lin, Fan-Yu Liao, and Yu-Hsuan Yu. 2023. Dynamic repositioning problem of dockless electric scooter sharing systems. *Transportation Letters* 15, 9 (2023), 1066–1082.

[4] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2020. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems* 21, 3 (2020), 1086–1095.

[5] Xiangxiang Chu and Hangjun Ye. 2017. Parameter Sharing Deep Deterministic Policy Gradient for Cooperative MARL. arXiv:1710.00336.

[6] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*. 2137–2145.

[7] Jayesh K. Gupta, Maxim Egorov, and Mykel J. Kochenderfer. 2017. Cooperative Multi-Agent Control Using Deep Reinforcement Learning. In *Adaptive Learning Agents Workshop @ AAMAS*.

[8] Suining He and Kang G. Shin. 2019. Spatio-Temporal Adaptive Pricing for Balancing Mobility-on-Demand Networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 4 (July 2019), 39:1–39:28.

[9] Suining He and Kang G. Shin. 2022. Spatio-Temporal Capsule-Based Reinforcement Learning for Mobility-on-Demand Coordination. *IEEE Transactions on Knowledge and Data Engineering* 34, 3 (Mar 2022), 1446–1461.

[10] Runqiu Hu, Zhizheng Zhang, Xinwei Ma, and Yuchuan Jin. 2021. Dynamic rebalancing optimization for bike-sharing system using priority-based MOEA/D algorithm. *IEEE Access* 9 (2021), 27067–27084.

[11] Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. 2022. The 37 Implementation Details of Proximal Policy Optimization. In *ICLR Blog Track*.

[12] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. 2021. Celebrating Diversity in Shared Multi-Agent Reinforcement Learning. In *Proc. NeurIPS*. 3991–4002.

[13] Hui Li, Xin Sun, Xueqian Wang, and Shaojie Shen. 2020. i-Rebalance: Personalized Vehicle Repositioning for Supply-Demand Balance. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. 480–487.

[14] Xinghua Li, Xinyuan Zhang, Cheng Cheng, Wei Wang, and Chao Yang. 2022. Dynamic Repositioning in Dock-Less Bike-Sharing System: A Multi-Agent Reinforcement Learning Approach. In *IEEE Intelligent Transportation Systems Conference (ITSC)*. 3352–3357.

[15] Yilin Liu, Guiyang Luo, Quan Yuan, Jinglin Li, Lei Jin, Bo Chen, and Rui Pan. 2023. GPLight: Grouped Multi-Agent Reinforcement Learning for Large-Scale Traffic Signal Control. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*. 172–180.

[16] Alessandro Staffolani, Victor-A. Darvariu, Maria A. Cheema, and Mirco Musolesi. 2025. A Cost-Aware Adaptive Bike Repositioning Agent Using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems* (2025).

[17] Justin K. Terry, Nathaniel Grammel, Sanghyun Son, Benjamin Black, and Aakriti Agrawal. 2020. Revisiting Parameter Sharing in Multi-Agent Deep Reinforcement Learning. arXiv:2005.13625.

[18] Jingwei Wang, Qianyue Hao, Wenzhen Huang, Xiaochen Fan, Zhentao Tang, Bin Wang, Jianye Hao, and Yong Li. 2024. DyPS: Dynamic Parameter Sharing in Multi-Agent Reinforcement Learning for Spatio-Temporal Resource Allocation. In *Proc. ACM KDD*. 3128–3139.

[19] Tonghan Wang, Jingyang Zhou, Qianchuan Zhao, and et al. 2021. RODE: Learning Roles to Decompose Multi-Agent Tasks. In *9th International Conference on Learning Representations (ICLR)*.

[20] Xi Yang, Suining He, and Mahan Tabatabaie. 2023. Equity-Aware Cross-Graph Reinforcement Learning for Bike Station Network Expansion. In *Proceedings of the 2023 ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. 45:1–45:12.

[21] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean Field Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 5571–5580.