Dynamic Mean-Field Control for Network MDPs with Exogenous Demand

Anonymous authors
Paper under double-blind review

Abstract

This paper studies the network control problems with exogenous demand, where network controller must dynamically allocate resources to satisfy exogenous demands with unknown distributions. We formalize the problem using Networked Markov Decision Processes with Exogenous Demands (Exo-NMDPs), where the system states are decoupled into endogenous states and stochastic exogenous demands. However, Exo-NMDPs pose three main challenges: scalability in large-scale networks; stochasticity from fluctuating exogenous demands; and delayed feedback of scheduling actions. To address these issues, we propose the Dynamic Mean-Field Control (DMFC) algorithm, a scalable and computationally efficient approach for matching exogenous demands. Specifically, DMFC transforms the high-dimensional actual states of the Exo-NMDP into low-dimensional mean-field states, and dynamically optimizes the policy by solving a mean-field control problem at each time step. This enables DMFC to capture spatiotemporal correlations between demand and system state, while remaining robust against demand fluctuations and action execution delay. We validate DMFC on two representative scenarios: supply-chain inventory management and vehicle routing. Our experimental results show that DMFC adapts well to various demand patterns and outperforms state-of-the-art baselines in both scenarios.

1 Introduction

Network control problems with exogenous demand has a broad application in real-world scenarios, including: supply chain management Bellamy & Basole (2013); Zhang et al. (2014); Aminzadegan et al. (2019), scheduling in robotic systems Rus; (2012); Pavone; (2016), and vehicle routing in mobility-on-demand systems Bullo et al. (2011); Holler et al. (2019); Gammelli et al. (2021). The control policies are the operational backbone of these systems, enhancing service reliability and driving cost reduction through dynamic and adaptive agent scheduling. However, designing such policies is challenging due to: 1) scalability in large-scale networks, 2) stochasticity from fluctuating exogenous demand, and 3) delayed feedback of control actions. These challenges lead to spatiotemporal mismatches between agents and demand, requiring a control policy π that adapts to evolving demand while incorporating past decisions for effective coordination.

We formulate network control problems with exogenous demand as a Networked Markov Decision Processes with Exogenous Demands (Exo-NMDPs), extending Exo-MDPs Sinclair et al. (2023) into networking settings. To address Exo-NMDPs, we introduce a mean-field Exo-NMDP formulation to transform the high-dimensional actual state into low-dimensional mean-field state, capturing both the endogenous system dynamics and the exogenous demand signals. Based on this formulation, we develop the Dynamic Mean-Field Control (DMFC) framework, which operates in two stages at each time step t: First, DMFC constructs an ideal mean-field state by incorporating predicted future demand into the current endogenous system state. Second, it solves a linear program that yields the control policy π_t by optimizing the system objective subject to mean-field dynamics and constraints.

The major contributions of our paper are listed below.

- **Problem Formulation.** We model the network control problems with exogenous demand as a networked Markov decision process with exogenous demands (Exo-NMDP), where the system states are decoupled into endogenous states and stochastic exogenous demands. The decomposition enables a more efficient and concise representation for policy design.
- **Algorithm Design.** We propose Dynamic Mean-Field Control (DMFC) algorithm for Exo-NMDP. DMFC leverages historical information to infer ideal mean-field states and then synthesizes a control policy to align system states towards the target states. This addresses "the curse of networked agents" and intricate spatio-temporal correlations.
- **Experimental Evaluation.** We evaluate DMFC in two real-world scenarios: supply chain inventory management and vehicle routing in mobility-on-demand systems. Experimental results show that our algorithm outperforms the state-of-the-art baseline in both applications. Our code is available at `https://anonymous.4open.science/r/DMFC-1247-F3E4T38D/`.

2 Related Work

Network control problems with exogenous demand have been studied from three main perspectives: RL-based (Reinforcement Learning), MPC-based (Model Predictive Control), and queueing-based methods. Here, we provide an overview of each approach. As mobility-on-demand systems are the representative example, we include them in the discussion.

RL-based methods: (Deep) reinforcement learning methods Sutton & Barto (2018); Mnih et al. (2013); Ladosz et al. (2022) offers promising solutions for network control problems with exogenous demand, particularly in mobility-on-demand systems Qin et al. (2022); Wen et al. (2024). Recent hybrid approaches integrate RL with optimization: Wei et al. (2024) combine the mean-field optimization formulation in Braverman et al. (2019) with temporal difference (TD) learning, but relies heavily on historical demand trajectories; Gammelli et al. (2023) develop a Graph RL framework under bi-level optimization that first predicts reward-driven desired next states then solves for policy from linear programs, yet struggles with scalability and generality in complex networks. Beyond single-agent paradigms, multi-agent RL (MARL) approaches encounter inherent dimensionality challenges. For instance, Lin et al. (2019) models vehicles as discrete-action agents, where complexity grow exponentially with agent counts. To address scalability limitations, Liu et al. (2022) employs regional agents while Wang et al. (2024) implements dynamic parameter sharing. However, communication and coordination between agents remain unsolved. These challenges motivate mean-field control (MFC) Gast et al. (2012); Bäuerle (2023) and mean-field RL (MFRL) Carmona et al. (2023); Pásztor et al. (2023); Jusup et al. (2024), which simplify multi-agent interactions through representative agents operating within aggregated distributions. A practical demonstration comes from Jusup et al. (2025), who apply MFC and MFRL to vehicle rebalancing via fleet dynamics modeling to avoid direct vehicle-to-vehicle coordination. However, their framework assumes instantaneous task completion, which is a critical limitation given operational delays observed in physical systems.

MPC-based methods: Model Predictive Control (MPC) Kouvaritakis & Cannon (2016); Borrelli et al. (2017) has become predominant in network control problems with exogenous demand, particularly in vehicle repositioning scenarios Iglesias et al. (2018); Tsao et al. (2019); Aalipour & Khani (2024). However, MPC performance critically depends on model accuracy, and developing a high-fidelity model is both time and data-intensive. Furthermore, as the look-ahead steps increase, the computational complexity of MPC also increases, especially for a large-scale network with substantial agents.

Queueing-based methods: Queueing theory Shortle et al. (2018) provides a framework for network resource allocation under exogenous demand constraints. Prior works Iglesias et al. (2016); Banerjee et al. (2022) investigate the problem within a queueing network, deriving routing policies by analyzing the stationary distribution of a Markov chain. One of the most closely related studies in this area is Braverman et al. (2019), which formulates the problem as a BCMP queueing network Baskett et al. (1975) and employs a mean-field optimization approach to determine the optimal mean-field control policy. However, these studies primarily focus on the steady-state solutions under static environments, lacking adaptability to non-stationary or time-varying demands, where system states state evolves over time.

We proposes the mean-field Exo-NMDP framework to model network control problems with exogenous demand, and introduce DMFC, a scalable and robust algorithm that coordinates agents via mean-field control while explicitly accounting for operational delays. Experiments show that DMFC offers superior adaptability to dynamic demand variations and outperforms the state-of-the-art methods in supply chain inventory management and vehicle routing in mobility-on-demand systems.

3 Problem Formulation

In this paper, we study a large-scale network control problem with exogenous demand over a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges. At each time step, the system makes decisions to dispatch agents from node u to node v for optimizing exogenous demand satisfaction. We formulate the problem as a Networked Markov Decision Process with Exogenous Demand.

3.1 Networked MDPs with Exogenous Demands (Exo-NMDPs)

We propose Networked Markov Decision Processes with Exogenous Demand (Exo-NMDPs), which extend Exo-MDPs Sinclair et al. (2023) to multi-agent systems where agents are dynamically scheduled in response to exogenous demand. An Exo-NMDP is defined by the tuple $(\mathcal{N}, \mathcal{G}, \mathcal{S}, \Xi, \mathcal{A}, \mathcal{P}_{\mathcal{S}}, \mathcal{P}_{\Xi}, \mathcal{R}, \gamma)$. Here, $\mathcal{N} = \{1, 2, \cdots, n\}$ is the set of agents operating on network topology $\mathcal{G}$. The endogenous state space is $\mathcal{S} := \prod_{k \in \mathcal{N}} \mathcal{S}_k$, where each agent k has a local state $s_{k,t} \in \mathcal{S}_k$. The exogenous demand process is modeled as a stochastic sequence $\Xi := \{\xi_t\}_{t \geq 0}$, where each $\xi_t = \{\xi_{v,t}\}_{v \in \mathcal{V}} \subset \mathbb{N}_+^{|\mathcal{V}|}$ represents node-specific, time-varying demands. In our setting, ξ_t is revealed at the start of time t, as opposed to Sinclair et al. (2023), where it becomes known only after actions are taken. Given ξ_t, the system selects joint actions $a_t = (a_{1,t}, \cdots, a_{n,t}) \in \mathcal{A} := \prod_{k \in \mathcal{N}} \mathcal{A}_k$ via a policy $\pi_t : \mathcal{S} \times \Xi^t \to \Delta(\mathcal{A})$, which maps current state s_t and demand history $\xi_{1:t}$ to a distribution over actions. The system endogenous state evolves according to $s_{t+1} \sim \mathcal{P}_{\mathcal{S}}(\cdot | s_t, a_t, \xi_t)$, while the exogenous demand follows a stochastic process $\xi_{t+1} \sim \mathcal{P}_{\Xi}(\cdot | \xi_{1:t})$. The reward function $\mathcal{R} : \mathcal{S} \times \Xi \times \mathcal{A} \to \mathbb{R}$ evaluates system performance at each time step, for example, based on revenue from satisfied demand. Future rewards are discounted by a factor $\gamma \in [0, 1)$. We assume that both $\mathcal{P}_{\mathcal{S}}$ and $\mathcal{R}$ are known, and that uncertainty arises only from $\mathcal{P}_{\Xi}$. The objective is to learn a policy that maximizes the expected long-term cumulative reward.

Exo-NMDPs as Multi-agent Semi-Markov Decision Processes. In Exo-NMDPs, agent actions may involve delays, such as travel time between network nodes. To model these temporal extensions, we adopt the option framework Sutton et al. (1999), where each option represents a temporally extended action that may span multiple time steps. For each individual agent, the option framework transforms its local MDP into a semi-Markov decision process (SMDP) Ross (1992), as the original action set $\mathcal{A}$ is replaced by a fixed set of options $\mathcal{O}$ (without loss of generality, in the subsequent discussion, we will abuse the term “actions” to also refer to “options”). While each agent operates under its own SMDP, the overall Exo-NMDP cannot be reduced to a single SMDP due to the asynchronous nature of agent decisions and variable option durations. Instead, we model the system as a Multiagent SMDP (MSMDP) Ghavamzadeh & Mahadevan (2004), where decision epochs are aligned with fixed-length time intervals (e.g., every 5 minutes). At each decision epoch t, only the subset of agents whose previous actions have just completed make new decisions, while the remaining agents continue executing their current actions. The MSMDP formulation of Exo-NMDPs enables tractable analysis across agents; however, the challenges of multi-agent scalability and communication remain unresolved.

Mean-Field Representation. To address the scalability and communication limitations inherent in Exo-NMDPs, we employ a mean-field formulation that aggregates individual agent behaviors into node-level dynamics, treating network nodes as the primary decision units. Let $\ell_{k,t} \in \mathcal{L}$ denote the location of agent k at time t, which is embedded in its full state $s_{k,t} := (\ell_{k,t}, \cdots)$. The system mean-field state is defined as: $\boldsymbol{\mu}_t = \{\boldsymbol{\mu}_t^v(i)\}_{i\in\mathcal{V}} \cup \{\boldsymbol{\mu}_t^e(u,v)\}_{(u,v)\in\mathcal{E}}$, where $\boldsymbol{\mu}_t^v(i) := \frac{1}{n}\sum_{k=1}^n \mathbb{I}(\ell_{k,t} = v_i)$ denotes the fraction of agents currently idle at node i, and $\boldsymbol{\mu}_t^e(u,v) := \frac{1}{n}\sum_{k=1}^n \mathbb{I}(\ell_{k,t} = e_{uv})$ captures the fraction of agents in transit along edge $e_{uv} \in \mathcal{E}$. The former corresponds to agents available for new decisions at time t, while the latter represents those with ongoing actions. The exogenous demand is similarly normalized as $\boldsymbol{\xi}_t := \{\lambda_{v,t}, \Phi_{vv',t}\}$, where $\lambda_{v,t} := \frac{1}{n}\xi_{v,t}$ reflects the demand intensity at node v, and $\Phi_{vv',t}\} \in \Delta(\mathcal{V})$ is the empirical conditional distribution over destinations v' given an origin v, estimated from observed demands. When destination information is unavailable or unnecessary—such as in inventory systems where demand does not induce agent relocation and only local stock levels of nodes matter—Φ_t can be omitted. Based on this representation, joint actions a_t are sampled from a mean-field policy $\pi_t(a_t \mid \boldsymbol{\mu}_t, \boldsymbol{\xi}_{1:t})$, which maps the current mean-field state and demand history to control decisions for each node. This abstraction transforms the system high-dimensional actual state into low-dimensional mean-field state, which greatly simplifies the system as the number of agent grows.

3.2 Networked Control Problems with Exogenous Demand

We formulate the networked control problems with exogenous demand as a mean-field Exo-NMDP and further adopt a fluid modeling perspective (similar to the approach in Braverman et al. (2019)) to characterize the system macroscopic evolution through flow dynamics. Here, $\boldsymbol{\mu}_t^v(i)$ denotes the idle agent density (or stock) at node i, $\boldsymbol{\mu}_t^e(u,v)$ captures the in-transit flow of agents along edge (u,v), while actions a_t represent newly initiated outflows. This fluid abstraction yields a tractable, low-dimensional system that supports scalable and robust control policy design.

Flow-Based Mean-Field Evolution. Two types of flows drive system evolution: demand flow and reposition flow. The demand flow from node i is defined as $f_{i,t}^D = \sum_j f_{ij,t}^D = \sum_j \alpha_{i,t}\lambda_{i,t}\Phi_{ij,t}$, where $\alpha_{i,t} \in [0,1]$ is the fulfillment rate, $\lambda_{i,t}$ is the normalized demand intensity, and $\Phi_{ij,t}$ is the conditional probability distribution of demand from i being routed to j. Both $\lambda_{i,t}$ and $\Phi_{ij,t}$ are components of $\boldsymbol{\xi}_t$. Depending on the system, fulfilled demand may either trigger agent relocation (e.g., in ride-sharing or delivery tasks), or remove agents from the system entirely (e.g., in inventory consumption). The reposition flow describes proactive agent movement based on a routing policy to balance supply and demand. It is defined as $f_{i,t}^R = \sum_j f_{ij,t}^R = \sum_j q_{ij,t} \cdot \boldsymbol{\mu}_t^v(i)$, where $\boldsymbol{\mu}_t^v(i)$ is the current density of idle agents at node i, and $q_{ij,t} \in \Delta(\mathcal{V})$ is the routing policy that specifies the probability of routing an idle agent from node i to j.

Our framework naturally extends to settings where agents have multiple types or internal states, indexed by a finite set $\mathcal{K}$. In this case, the mean-field state at each node i is now a vector $\boldsymbol{\mu}_t^v(i) \in \mathbb{R}^{\mathcal{K}}$ where each element tracks the density of agents of that type at node i. To capture internal transitions between types (e.g., status changes), we define a conversion flow $f_{i,t}^C = C_{i,t} \cdot \boldsymbol{\mu}_t^v(i)$, where $C_{i,t}$ is the type transition matrix at node i.

In practice, flows f^D, f^R, and f^C often incur delays. In other words, both agent type transitions and repositioning require some time, which we denote by a delay parameter τ. We incorporate this into a flow conservation model as follows:

$$\max \quad \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(\boldsymbol{\mu}_t, \boldsymbol{\xi}_t, a_t)\right] \tag{1}$$

$$\text{s.t.} \quad f_{i,t}^{out} = f_{i,t}^D + f_{i,t}^R + f_{i,t}^C \tag{2}$$

$$f_{i,t}^{in} = \sum_k (f_{ki,t-\tau}^D + f_{ki,t-\tau}^R) + C_{i,t-\tau} f_{i,t-\tau}^C \tag{3}$$

$$\boldsymbol{\mu}_{i,t+1} = \boldsymbol{\mu}_{i,t} + f_{i,t}^{in} - f_{i,t}^{out} \tag{4}$$

At each step t, we assume that demand-driven flows f^D take place first, then agent repositioning f^R, and finally local conversion f^C. Equation (2) represents the total outflow and Equation (3) captures the delayed inflow. The system's mean-field state evolves according to the flow conservation dynamics in Equation (4). The objective (1) tries to maximize system long-term revenue.

4 Method

The stochastic nature of exogenous demand renders direct solutions to (1)–(4) intractable. To address this, we propose the Dynamic Mean-Field Control (DMFC) framework for stepwise policy optimization. At each time t, DMFC first predicts a ideal mean-field state $\boldsymbol{\mu}_t^*$ with historical information, and then solves for the policy π_t from a mean-field control problem (6) - (7) via linear programming. The overall closed-loop interaction of DMFC with the environment is presented in Algorithm 1.

Algorithm 1 Dynamic Mean-Field Control Loop with Environment Interaction

Require: network $\mathcal{G}$, time horizon T

1: **for** $t = 1, \cdots, T$ **do**
2: **Receive** inflows f_t^{in} from external sources or previous steps (environment update)
3: **Observe** endogenous system state s_t and exogenous demand ξ_t
4: **Scale** to the mean-field level: compute $\boldsymbol{\mu}_t$ and demand $\boldsymbol{\xi}_t = (\boldsymbol{\lambda}_t, \Phi_t)$
5: **Match** supply ($\boldsymbol{\mu}_t$) with demand ($\boldsymbol{\xi}_t$) for demand flow f_t^D
6: **Forecast** future demand $\hat{\boldsymbol{\xi}}_{t+1}$ based on historical observations $\boldsymbol{\xi}_{1:t}$
7: **Predict** future supply $\hat{\boldsymbol{\mu}}_{t+1}$ by aggregating current state and delayed inflows
8: **Construct** the ideal mean-field state $\hat{\boldsymbol{\mu}}_{t+1}^*$ by demand-proportional allocation
9: **Solve** the mean-field control problem (6)–(7) with ideal mean-field state $\hat{\boldsymbol{\mu}}_{t+1}^*$ to obtain π_t
10: **Execute** policy π_t in the environment to generate flows f_t^R and f_t^C
11: **end for**

4.1 The Ideal Mean-Field State

The ideal mean-field state $\hat{\boldsymbol{\mu}}_{t+1}^*$ serves as a data-driven intermediate state that guides the stepwise policy optimization, which is constructed through the following steps. First, DMFC forecasts next-step demand $\hat{\boldsymbol{\xi}}_{t+1}$ using a weighted history: $\hat{\boldsymbol{\xi}}_{t+1} = \mathbf{w}_{1:t} \cdot \boldsymbol{\xi}_{1:t}$, where $\mathbf{w}_{1:t}$ serves as learnable weight parameters controlling the influence of past observations $\boldsymbol{\xi}_{1:t}$ on the prediction $\hat{\boldsymbol{\xi}}_{t+1}$. To account for agent latency, projected future agent availability is estimated as $\hat{\boldsymbol{\mu}}_{i,t+1} = \boldsymbol{\mu}_{i,t} + \sum_{\tau=t+1}^{t+\Delta t} f_{i,\tau}^{\text{in}}$. The ideal mean-field state $\hat{\boldsymbol{\mu}}_{t+1}^*$ is then constructed by demand-proportional allocation:

$$\hat{\boldsymbol{\mu}}_{i,t+1}^* = \frac{\hat{\boldsymbol{\xi}}_{i,t+1}}{\sum_k \hat{\boldsymbol{\xi}}_{k,t+1}} \sum_k \hat{\boldsymbol{\mu}}_{k,t+1}, \quad \forall i \in \mathcal{V} \tag{5}$$

The ideal mean-field state $\boldsymbol{\mu}_{i,t}^*$ improves both short and long-term performance by balancing supply with demand. In the short term, the proportional allocation scheme ensures high demand fulfillment and immediate reward maximization. In the long term, despite flow delays, proportional allocation allows the system to gradually concentrate supply where demand is high, enhancing robustness and sustaining long-term efficiency.

4.2 Dynamic Mean-Field Control

With the ideal mean-field state $\hat{\boldsymbol{\mu}}_{t+1}^*$ defined, we seek a control policy π_t that moves the system from the its current state $\boldsymbol{\mu}_t$ toward this target by solving:

$$\max_{a_t} \quad \mathbb{E}\left[\mathcal{R}(\boldsymbol{\mu}_t, \boldsymbol{\xi}_t, a_t) - \mathbf{d}(\boldsymbol{\mu}_{t+1}, \hat{\boldsymbol{\mu}}_{t+1}^*)\right] \tag{6}$$

$$\text{s.t.} \quad \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + f_t^{in} - f_t^{out} \tag{7}$$

Here, the reward function $\mathcal{R}(\boldsymbol{\mu}_t, \boldsymbol{\xi}_t, a_t)$ evaluates immediate performance through demand fulfillment, while the discrepancy term $\mathbf{d}(\boldsymbol{m}'_t, \boldsymbol{\mu}^*_t)$ penalizes deviations from the forecasted ideal state. The constraint in (7) models the system dynamics via flow conservation. In practice, however, due to stochastic disturbances, delayed flows, and model approximation errors, the policy π_t may not perfectly align the evolved system state $\boldsymbol{\mu}_{t+1}$ with the ideal state $\hat{\boldsymbol{\mu}}^*_{t+1}$. Instead, the system tends to stabilize around the ideal trajectory, maintaining a dynamic equilibrium that supports robust performance across varying conditions.

5 Experimental evaluation

In this section, we implemented DMFC and conducted experiments in two representative scenarios: (1) **supply chain inventory management**, following the experimental setup of the GRL framework Gammelli et al. (2023); and (2) **vehicle routing in mobility-on-demand systems**, where we adopt the original GRL configuration and further introduce augmented datasets with increased demand variability.

5.1 Supply Chain Inventory Management (SCIM)

We consider the Supply Chain Inventory Management (SCIM) problem as a mean-field Exo-NMDP, enabling scalable optimization of commodity production, transportation, and inventory control under uncertainty. In this framework, the supply chain network is modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, compromising factories ($\mathcal{V}_F$) and stores ($\mathcal{V}_S$), where each node acts as a control agent managing aggregate flows.

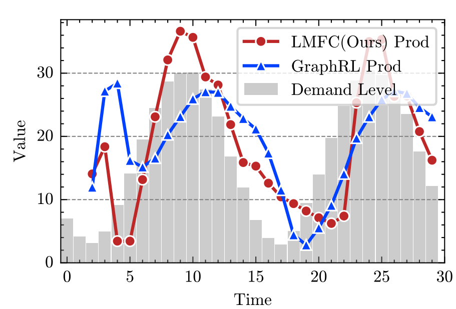
States, Demands, Actions, and Rewards. The system mean-field state $\boldsymbol{\mu}_t$ captures the normalized inventory levels at nodes and in-transit flow on edges. At each time t, exogenous demand $\boldsymbol{\xi}_t$ arrives at store nodes $v \in \mathcal{V}_S$. If local inventory is available, demand is fulfilled with revenue p. Otherwise it remains pending and accumulates a delay penalty ϵ per time step. Each node select actions over feasible flow decisions. Factories control commodity production (conversion flow f_t^C) ant outbound shipments (reposition flow f_t^R), incurring production cost m^P, transportation cost m^T, and delays t^P, t_{ij}. Stores passively fulfill demand (demand flow f_t^D) and accept incoming shipments. All nodes face storage constraints c_i, incur storage cost m^S, and are penalized by ϵ for overstocking. The policy $\pi_t(a_t|\boldsymbol{\mu}_t, \boldsymbol{\xi}_{1:t})$ maps currently observed inventory level $\boldsymbol{\mu}_t$ and demand history $\boldsymbol{\xi}_{1:t}$ to node-level flow actions. The reward function of the SCIM system is the total profit of the system, calculated as total revenue minus operational costs and penalties.

	DMFC	GRL	Improvement
1F2S	432 ± 193*	247 ± 110	+75%
1F3S	1671 ± 223*	875 ± 102	+91%
1F10S	4310 ± 358*	1244 ± 312	+246%

(a) Reward improvement of DMFC over GraphRL.

	DMFC	GRL	Reduction
1F2S	766*	1113	-31.2%
1F3S	676*	1227	-44.9%
1F10S	2278*	5080	-55.2%

(b) Storage cost reduction of DMFC over GraphRL.



(c) Production-demand comparison on 1F3S dataset: DMFC vs. GraphRL.

Figure 1: Performance comparison between DMFC algorithm and GraphRL baseline

Supply Chain Inventory Control. We apply the DMFC policy to the SCIM problem and compare it against the GraphRL baseline. All results are averaged over 30 runs, with detailed settings provided in Appendix A.1.1. Tables 1a and 1b show that DMFC consistently outperforms GraphRL across all benchmark scenarios (1F2S, 1F3S, and 1F10S; F = factory, S = store). Specifically, DMFC improves total reward by 75–246% while reducing storage costs by 31–55%, without increasing penalties or degrading fulfillment performance. Figure 1c further illustrates DMFC's advantage in production planning. On the 1F3S dataset, DMFC achieves better alignment between production and demand trends, resulting in smoother inventory turnover and lower storage costs. Results confirm that the mean-field modeling and lookahead scheduling enable more efficient resource utilization and improved responsiveness to exogenous demand fluctuations.

5.2 Vehicle Routing in Mobility-on-Demand Systems

Vehicle routing problems in mobility-on-demand systems (ECR) can also be formualted as mean-field Exo-NMDPs, where the goal is to optimize the repositioning of idle vehicles in a transportation network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to maximize passenger demand fulfillment. Here, network nodes denote geographic regions and edges capture feasible inter-region routes. Each region is considered a central controller agent that governs the allocation and repositioning of idle vehicles within its area.

States, Demands, Actions, and Rewards. The mean-field state $\boldsymbol{\mu}_t = (\boldsymbol{\mu}_t^v(i), \boldsymbol{\mu}_t^e(i,j))$ describes the normalized distribution of idle vehicles at each region $i \in \mathcal{V}$ and in-transit vehicles along each edge $(i,j) \in \mathcal{E}$. Exogenous demand is given by $\boldsymbol{\xi}_t := (\lambda_t(v), \Phi_t(v, v'))$, where $\lambda_t(v)$ is the normalized arrival rate of passenger requests at region v, and $\Phi_t(v, v')$ represents the empirical distribution over destinations. When passengers arrive in region v, they are instantly matched to idle vehicles in the same region, generating a demand flow f^D. If supply falls short in that region, unmatched passengers leave the system. To reduce future imbalances, regions execute reposition actions, giving rise to the reposition flow f^R. Both f^D and f^R are subject to delays t_{ij} due to travel time. Upon arrival, vehicles automatically transition back to idle state, generating the conversion flow f^C. The reward is defined as total fare revenue minus fuel costs, and the objective is to optimize repositioning to maximize long-term returns.

Empty Car Coordination. We evaluate DMFC on four benchmarks (New York, Shenzhen, DiDi-9, and DiDi-20) and compare it against the GraphRL baseline. All results are averaged over 30 runs, with each episode consisting of 200 steps. The evaluation metric is the Order Response Rate (ORR) Lin et al. (2019), defined as the ratio of served requests to total requests. Detailed experimental settings can be found in Appendix A.2.1. Figure 2 illustrates the demand patterns for each dataset. To ensure consistent evaluation, all datasets have been adjusted to span 200 time slots. The red line represents the average passenger demand across regions, while the shaded area indicates the regional demand heterogeneity. Clear periodic trends are observed in the (extended) New York and Shenzhen datasets, whereas distinct diurnal patterns are evident in the DiDi-9 and DiDi-20 datasets. See Appendix A.2.2 for more information on the datasets.

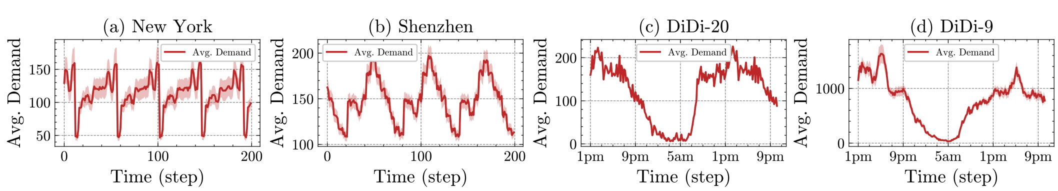
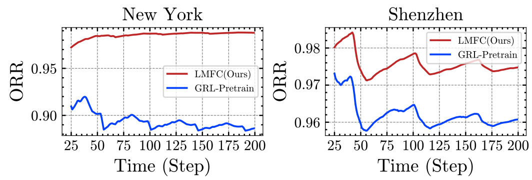
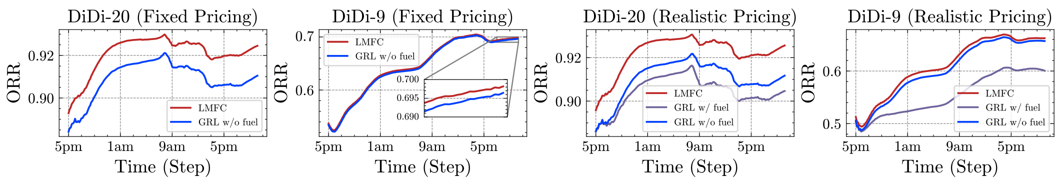


Figure 2: Demand pattern for each dataset in ECR environment

Figure 3a shows the ORR (Order Response Rate) performance on our (extended) New York and Shenzhen datasets with periodic demand. The GraphRL policy (GRL-Pretrain), which uses pre-trained weights directly from the original codebase without retraining, exhibits a clear performance degradation and limited adaptability across both cities. In contrast, DMFC maintains high and stable performance, effectively adapting to dynamic demand patterns in non-stationary environments. Figures 3b report the ORR under two pricing schemes: (i) fixed pricing (unit price, no fuel cost), and (ii) realistic pricing with or without fuel cost. DMFC consistently outperforms GraphRL across



(a) Performance on extended New York and Shenzhen datasets with periodic demand.



(b) Performance on DiDi-9 and DiDi-20 datasets under fixed and realistic pricing.

Figure 3: Order response rate (ORR) comparison across datasets and pricing schemes

all settings. Under fixed pricing, DMFC demonstrates a clear advantage on the DiDi-20 dataset and moderate improvements on DiDi-9. This is because DiDi-9 exhibits a more severe mismatch between supply and demand, where even marginal enhancements in routing yield noticeable benefits. The performance gap further widens under realistic pricing. While GraphRL tends to pursue short-term gains, often neglecting long-term system efficiency, DMFC emphasizes demand satisfaction, leading to more effective resource allocation and more stable performance under dynamic pricing and cost structures.

6 Conclusion

This paper proposes the Dynamic Mean-Field Control (DMFC) framework for optimizing resource allocation in network control problems with exogenous demands. Evaluations on supply chain inventory management and vehicle routing tasks demonstrate DMFC's superiority over the GraphRL baseline, with enhanced adaptability to demand fluctuations and network complexity. Further, the framework achieves generalizability across diverse demand patterns (periodic, diurnal, sparse), robustness against supply-demand imbalances, and scalability to large networks via linear computational complexity.

References

Ali Aalipour and Alireza Khani. Modeling, analysis, and control of autonomous mobility-on-demand systems: A discrete-time linear dynamical system and a model predictive control approach. *IEEE Transactions on Intelligent Transportation Systems*, 25(8):8615–8628, 2024. DOI: 10.1109/TITS.2024.3398562.

Sajede Aminzadegan, Mohammad Tamannaei, and Morteza Rasti-Barzoki. Multi-agent supply chain scheduling problem by considering resource allocation and transportation. *Computers & Industrial Engineering*, 137:106003, 2019. ISSN 0360-8352. DOI: 10.1016/j.cie.2019.106003.

Siddhartha Banerjee, Daniel Freund, and Thodoris Lykouris. Pricing and optimization in shared vehicle systems: An approximation framework. *Operations Research*, 70(3):1783–1805, 2022. DOI: 10.1287/opre.2021.2165.

Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *J. ACM*, 22(2):248–260, April 1975. ISSN 0004-5411. DOI: 10.1145/321879.321887.

Marcus Bellamy and Rahul Basole. Network analysis of supply chain systems: A systematic review and future research. *Systems Engineering*, 16, 06 2013. DOI: 10.1002/sys.21238.

Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

Anton Braverman, J. G. Dai, Xin Liu, and Lei Ying. Empty-car routing in ridesharing systems. *Operations Research*, 67(5):1437–1452, 2019. DOI: 10.1287/opre.2018.1822.

Francesco Bullo, Emilio Frazzoli, Marco Pavone, Ketan Savla, and Stephen L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482–1504, 2011. DOI: 10.1109/JPROC.2011.2158181.

Nicole Bäuerle. Mean field markov decision processes. *Applied Mathematics & Optimization*, 88, 04 2023. DOI: 10.1007/s00245-023-09985-1.

René Carmona, Mathieu Laurière, and Zongjun Tan. Model-free mean-field reinforcement learning: Mean-field MDP and mean-field Q-learning. *The Annals of Applied Probability*, 33(6B): 5334 – 5381, 2023. DOI: 10.1214/23-AAP1949. URL https://doi.org/10.1214/23-AAP1949.

DRI. http://research.xiaojukeji.com/index_en.html, 2016. Accessed: 2016-06-30.

Daniele Gammelli, Kaidi Yang, James Harrison, Filipe Rodrigues, Francisco C. Pereira, and Marco Pavone. Graph neural network reinforcement learning for autonomous mobility-on-demand systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2996–3003, 2021. DOI: 10.1109/CDC45484.2021.9683135.

Daniele Gammelli, James Harrison, Kaidi Yang, Marco Pavone, Filipe Rodrigues, and Francisco C. Pereira. Graph reinforcement learning for network control via bi-level optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10587–10610. PMLR, 23–29 Jul 2023.

Nicolas Gast, Bruno Gaujal, and Jean-Yves Le Boudec. Mean field for markov decision processes: From discrete to continuous optimization. *IEEE Transactions on Automatic Control*, 57(9):2266–2280, 2012. DOI: 10.1109/TAC.2012.2186176.

Mohammad Ghavamzadeh and Sridhar Mahadevan. Learning to communicate and act using hierarchical reinforcement learning. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1114–1121. Citeseer, 2004.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL https://www.gurobi.com.

John Holler, Risto Vuorio, Zhiwei Qin, Xiaocheng Tang, Yan Jiao, Tiancheng Jin, Satinder Singh, Chenxi Wang, and Jieping Ye. Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1090–1095, 2019. DOI: 10.1109/ICDM.2019.00129.

Ramon Iglesias, Federico Rossi, Rick Zhang, and Marco Pavone. A bcmp network approach to modeling and controlling autonomous mobility-on-demand systems. *The International Journal of Robotics Research*, 38, 07 2016. DOI: 10.1177/0278364918780335.

Ramon Iglesias, Federico Rossi, Kevin Wang, David Hallac, Jure Leskovec, and Marco Pavone. Data-driven model predictive control of autonomous mobility-on-demand systems. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6019–6025, 2018. DOI: 10.1109/ICRA.2018.8460966.

Matej Jusup, Barna Pásztor, Tadeusz Janik, Kenan Zhang, Francesco Corman, Andreas Krause, and Ilija Bogunovic. Safe model-based multi-agent mean-field reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pp. 973–982, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Matej Jusup, Kenan Zhang, Zhiyuan Hu, Barna Pásztor, Andreas Krause, and Francesco Corman. Ride-sourcing vehicle rebalancing with service accessibility guarantees via constrained mean-field reinforcement learning. *CoRR*, abs/2503.24183, 2025. DOI: 10.48550/ARXIV.2503.24183. URL https://doi.org/10.48550/arXiv.2503.24183.

KDDCup. https://www.biendata.xyz/competition/kdd_didi/, 2020. Accessed: 2020-07-12.

Basil Kouvaritakis and Mark Cannon. *Model Predictive Control: Classical, Robust and Stochastic*. Springer, 2016.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Inf. Fusion*, 85:1–22, 2022. DOI: 10.1016/J.INFFUS.2022.03.003. URL https://doi.org/10.1016/j.inffus.2022.03.003.

Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. Efficient collaborative multi-agent deep reinforcement learning for large-scale fleet management, 2019. URL https://arxiv.org/abs/1802.06444.

Chenxi Liu, Chao-Xiong Chen, and Chao Chen. Meta: A city-wide taxi repositioning framework based on multi-agent reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13890–13895, 2022. DOI: 10.1109/TITS.2021.3096226.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019.

Barna Pásztor, Andreas Krause, and Ilija Bogunovic. Efficient model-based multi-agent mean-field reinforcement learning. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=gvcDSDYUZx.

Rick Zhang;Marco Pavone;. Control of robotic mobility-on-demand systems: A queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1-3):186–203, 2016. DOI: 10.1177/0278364915581863.

Zhiwei Tony Qin, Hongtu Zhu, and Jieping Ye. Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144:103852, 2022.

Sheldon M Ross. *Applied probability models with optimization applications*. Dover Publications, 1992.

Marco Pavone;Stephen L Smith;Emilio Frazzoli;Daniela Rus;. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31(7):839–854, 2012. DOI: 10.1177/0278364912444766.

John F Shortle, James M Thompson, Donald Gross, and Carl M Harris. *Fundamentals of queueing theory*. John Wiley & Sons, 2018.

Sean R. Sinclair, Felipe Frujeri, Ching-An Cheng, Luke Marshall, Hugo Barbalho, Jingling Li, Jennifer Neville, Ishai Menache, and Adith Swaminathan. Hindsight learning for mdps with exogenous inputs, 2023. URL https://arxiv.org/abs/2207.06272.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999. ISSN 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(99)00052-1. URL https://www.sciencedirect.com/science/article/pii/S0004370299000521.

Matthew Tsao, Dejan Milojevic, Claudio Ruch, Mauro Salazar, Emilio Frazzoli, and Marco Pavone. Model predictive control of ride-sharing autonomous mobility-on-demand systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6665–6671, 2019. DOI: 10.1109/ICRA.2019.8794194.

Jingwei Wang, Qianyue Hao, Wenzhen Huang, Xiaochen Fan, Zhentao Tang, Bin Wang, Jianye Hao, and Yong Li. Dyps: Dynamic parameter sharing in multi-agent reinforcement learning for spatio-temporal resource allocation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 3128–3139, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901.

Honghao Wei, Zixian Yang, Xin Liu, Zhiwei Qin, Xiaocheng Tang, and Lei Ying. A reinforcement learning and prediction-based lookahead policy for vehicle repositioning in online ride-hailing systems. *IEEE Transactions on Intelligent Transportation Systems*, 25(2):1846–1856, 2024. DOI: 10.1109/TITS.2023.3312048.

Dacheng Wen, Yupeng Li, and Francis C. M. Lau. A survey of machine learning-based ride-hailing planning. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):4734–4753, 2024. DOI: 10.1109/TITS.2023.3345174.

Zhi-Hai Zhang, Bin-Feng Li, Xiang Qian, and Lin-Ning Cai. An integrated supply chain network design problem for bidirectional flows. *Expert Systems with Applications*, 41(9):4298–4308, 2014. ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.12.053.

A Additional Experiment Details

This section provides further details about the experimental configuration and hyperparameters. All RL modules are taken from Gammelli et al. (2023) and implemented using Pytorch Paszke et al. (2019). The Gurobi Optimizer Gurobi Optimization, LLC (2024) is used for optimization problems. The code environment has been rewritten to facilitate more detailed and adaptable customization options. To ensure consistent behavior across environments in edge cases, some IBM CPLEX components from the original codebase of Gammelli et al. (2023) have been retained within our framework.

A.1 Supply Chain Inventory Management

A.1.1 Environment Details and Datasets

We follow the basic setting of Gammelli et al. (2023) and define the exogenous demand pattern λ_t as a co-sinusoidal function with a stochastic component:

$$\lambda_{i,t} = \left\lfloor \frac{\lambda_i^{\max}}{2} \left(1 + \cos\left(\frac{4\pi(2i+t)}{T}\right) + \mathcal{U}(0, \lambda_i^{\mathrm{var}})\right)\right\rfloor \tag{8}$$

where $\lfloor \cdot \rfloor$ is the floor function, $\lambda_i^{\max}$ is the maximum demand value $\mathcal{U}(0, \lambda_i^{\mathrm{var}})$ is a uniform distribution on the interval $[0, \lambda_i^{\mathrm{var}}]$, and T is the episode length. Hyperparameters in simulation experiments are borrowed from Gammelli et al. (2023) and listed below (with minor modifications):

A.1.2 Mean-Field Control

In a supply chain inventory management problem, commodities are modeled as agents whose distributions evolve over time. At each time step t, the system state s_t comprises current inventory levels at factories and stores, along with pending demands at each store node that have accumulated due to prior stockouts. Given the system state and exogenous demand ξ_t, we define the mean-field state $\boldsymbol{\mu}_t$ and the demand vector $\boldsymbol{\xi}_t$, both normalized by the total number of commodities in the network.

The total commodity volume is not static, as items are continuously consumed (sold at stores) and replenished (produced by factories). To ensure a consistent mean-field representation, we incorporate demand-driven production into the scaling process. Specifically, we forecast the next-step demand, account for current inventories and backlog levels, and infer the appropriate production quantity to maintain system balance. This enables us to construct a normalized and dynamically adjusted mean-field state that reflects both supply and anticipated demand over time.

The mean-field control formulation of a SCIM problem is defined as follows:

$$\max_{f^R, f^C, \epsilon^V, \epsilon^s} \quad \min_{i \in \mathcal{V}_\mathcal{S}} \alpha_i - M \sum_{i \in \mathcal{V}} |\epsilon_i^V| - \sum_{i \in \mathcal{V}} |\epsilon_i^s| \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{\mu}_{i,t+1} = \boldsymbol{\mu}_{i,t} - f_{i,t}^D + \sum_{k \in \mathcal{V}_\mathcal{F}} f_{ki,t}^R, \quad \forall i \in \mathcal{V}_S \tag{10}$$

$$\boldsymbol{\mu}_{i,t+1} = \boldsymbol{\mu}_{i,t} - \sum_{j \in \mathcal{V}_\mathcal{S}} f_{ij,t}^R + f_{i,t}^C, \quad \forall i \in \mathcal{V}_F \tag{11}$$

$$\hat{\boldsymbol{\mu}}_{i,t+1}^* = \boldsymbol{\mu}_{i,t+1} + \epsilon_i^s, \quad \forall i \in \mathcal{V} \tag{12}$$

$$\boldsymbol{\mu}_{i,t+1} \le V_i + \epsilon_i^V, \quad \forall i \in \mathcal{V} \tag{13}$$

The objective (9) adopts a minimax structure, where $\min_{i \in \mathcal{V}_\mathcal{S}} \alpha_i$ denotes the worst-case demand fulfillment rate across all store nodes, analogous to the reward function $\mathcal{R}(\boldsymbol{\mu}_t, \boldsymbol{\xi}_t, a_t)$ in (6). The penalty terms $\sum_i |\epsilon^s i|$ and $M \sum i |\epsilon^V i|$ serve two purposes: the former captures the deviation between the evolved mean-field state $\boldsymbol{\mu} t + 1$ and the target state $\hat{\boldsymbol{\mu}}_{t+1}^*$ (as defined in (12)), while the latter penalizes inventory overflow beyond node capacity V_i, as constrained in (13).

Parameter	Explanation	Value
$\lambda^{\max}$	Maximum demand	$[2, 16]$
λ^{var}	Demand variance	$[2, 2]$
T	Episode length	30
t^P	Production time	1
t_{ij}	Travel time	$[1, 1]$
c	Storage capacity	$[20, 9, 12]$
m^P	Production cost	5
m^S	Storage cost	$[3, 2, 1]$
m^T	Transportation cost	$[0.3, 0.6]$
p	Price	15
ϵ	Penalty	21

(a) Parameters for the 1F2S environment

Parameter	Explanation	Value
$\lambda^{\max}$	Maximum demand	$[1, 5, 24]$
λ^{var}	Demand variance	$[2, 2, 2]$
T	Episode length	30
t^P	Production time	1
t_{ij}	Travel time	$[1, 1, 1]$
c	Storage capacity	$[30, 15, 15, 15]$
m^P	Production cost	5
m^S	Storage cost	$[2, 1, 1, 1]$
m^T	Transportation cost	$[0.3, 0.3, 0.3]$
p	Price	15
ϵ	Penalty	21

(b) Parameters for the 1F3S environment

Parameter	Explanation	Value
$\lambda^{\max}$	Maximum demand	$[2, 2, 2, 2, 10, 10, 10, 18, 18, 18]$
λ^{var}	Demand variance	$[2]_{i \in \mathcal{V}}$
T	Episode length	30
t^P	Production time	1
t_{ij}	Travel time	$[1]_{i \in \mathcal{V}}$
c	Storage capacity	$[100, 15 \quad \forall i \in \mathcal{V} \setminus 0]$
m^P	Production cost	5
m^S	Storage cost	$[1, 2 \quad \forall i \in \mathcal{V} \setminus 0]$
m^T	Transportation cost	$[0.3]_{i \in \mathcal{V}}$
p	Price	15
ϵ	Penalty	21

(c) Parameters for the 1F10S environment

Figure 4: Parameter settings for different SCIM environments

Constraints (10) and (11) refine the general flow conservation dynamics in (4) for the supply chain inventory management (SCIM) setting. Specifically, $f^D_{i,t}$ represents the quantity of demand fulfilled at store node i, $f^R_{ij,t}$ indicates the fraction of commodities transferred from factory i to store j, and $f^C_{i,t}$ denotes the amount of production at factory node i.

A.2 Vehicle Routing in Mobility-on-Demand Systems

A.2.1 Environment Details

We follow the experimental setup of Gammelli et al. (2023) and construct our environments accordingly. To ensure consistent and comprehensive evaluation, all experiments are conducted on episodes consisting of 200 time steps. We model vehicles within mobility-on-demand systems as agents. Exogenous demand is then modeled as a Poisson arrival process, with intensity specified for each time step. Agent repositioning and scheduling delays are determined by the datasets and may be either constant or time-varying. Note that the geographical regions in the environment are not necessarily fully connected.

A.2.2 Datasets

We conduct experiments on four benchmarks: the **New York** and **Shenzhen** datasets from Gammelli et al. (2023) (3-hour demand windows), the **DiDi-9** dataset from DRI (2016) used by Braverman et al. (2019) (21-day order records), and the **DiDi-20** dataset from KDDCup (2020) used by Wei et al. (2024) (a full-day order data). Table 1 gives an overview of each datasets. For preprocessing, we extended the New York and Shenzhen datasets to 200 timesteps. For the DiDi-9 and DiDi-20 datasets, we extracted contiguous 200-timestep segments spanning 1 PM to 10 PM across consecutive days. Figure 2 illustrates the preprocessed demand patterns, showing periodic trends in the New York/Shenzhen data and diurnal cycles in the DiDi-9/DiDi-20 datasets. The red line denotes average demand, with shading indicating regional variations.

	NYC	SZ	DiDi-9	DiDi-20
# of regions	14	17	9	20
# of cars	1200	1200	1500	500
minutes per step	4	3	10	10

Table 1: Configurations of ECR simulation experiments over all datasets

Each dataset contains comprehensive records of travel times, origin–destination pairs, pricing, and other relevant information. From these records, we extract key time-slot-specific parameters to model the dynamics of the traffic network, including the demand pattern ξ_t, inter-regional travel time t_{ij}, and the fuel cost coefficient β.

A.2.3 Mean-Field Control

When considering vehicle routing problems in a mobility-on-demand system, we consider the real-world as inter-connected geographical regions, and model each vehicle as an agent. At each time step t, the system mean-field state $\boldsymbol{\mu}_t$ represents the density of agents across each region. The ideal mean-field state of the system is defined in proportional to anticipated next-step demand of each region. We define the corresponding mean-field control problem as follows:

$$\min_{f^R} \quad \sum_{(i,j)\in\mathcal{E}} t_{ij} \cdot f_{ij}^R + M \cdot \epsilon_i \tag{14}$$

$$\text{s.t.} \quad \boldsymbol{\mu}_{i,t} + \sum_{k\neq i} f_{ki}^R - \sum_{j\neq i} f_{ij}^R + \epsilon_i \geq \hat{\boldsymbol{\mu}}_{i,t+1}^*, \quad \forall i \in \mathcal{V} \tag{15}$$

$$\sum_{j\neq i} f_{ij}^R \leq \boldsymbol{\mu}_{i,t} \quad \forall i \in \mathcal{V} \tag{16}$$

The objective in (14) seeks to minimize repositioning time, thereby reducing the fuel cost associated with vehicle routing. This objective is alternative to our overarching goal of maximizing long-term system revenue as defined in (1). The auxiliary variable ϵ captures deviations between the actual system mean-field state $\boldsymbol{\mu}_{t+1}$ and the target mean-field state $\hat{\boldsymbol{\mu}}_{t+1}^*$, with M denoting a large penalty coefficient. Constraint (15) ensures that the system evolves toward the desired target state, while constraint (16) enforces that the repositioning outflow from region i does not exceed the number of available empty vehicles.