Dynamic Courier Capacity Acquisition Model

Zhuang Kang

June 2025

Abstract

This document describes the dynamic courier capacity acquisition model from Auad et al. (2023), formulated as a Markov Decision Process (MDP) for a rapid delivery system. The model manages on-demand courier additions to balance service quality and costs under stochastic demand, focusing on the state, control, objective, and constraints without the learning phase.

1 Model Description

The dynamic courier capacity acquisition problem is modeled as a Markov Decision Process (MDP) to optimize courier capacity in a rapid delivery system over an operating period $[0, H]$, with $H = 540$ minutes. Orders are placed in the first $H_0 = 450$ minutes and must be delivered within 40 minutes of placement. Decisions to add on-demand couriers are made every $\Delta = 5$ minutes.

1.1 State

At each decision epoch $t \in T_{\text{action}}(\Delta) = \{i\Delta : i \in \{0, 1, \ldots, \lfloor H_0/\Delta \rfloor\}\}$, the state s_t comprises:

- $H - t$: Remaining operating time (seconds).
- q_t^{couriers}: Number of active couriers at time t.
- q_t^{orders}: Number of active orders (pending or assigned).
- Θ_t^1: Vector of scheduled courier additions and terminations in future epochs.
- Θ_t^2: Number of orders placed in the last 1800 seconds, normalized.
- Θ_t^3: Number of orders at risk of being late (due within 600 seconds).
- Θ_t^4: Average time to complete orders in Θ_t^3.

Two state variants are used: a 7-dimensional base state and a 21-dimensional state with additional order and courier dynamics.

1.2 Control

The control (action) $a_t = (a_t^1, a_t^{1.5})$ at time t specifies the number of on-demand couriers added for each courier type $c \in \mathcal{C} = \{3600, 5400\}$ seconds (1 or 1.5 hours), with $a_t^c \in \{0, 1, 2\}$. Added couriers start at $t + 300$ seconds at a pickup location with the highest unassigned order volume.

a_t^c is the number of couriers of type c (either 1-hour or 1.5-hour) added to the system at decision epoch t. The possible values $a_t^c \in \{0, 1, 2\}$ indicate:

- **0**: No couriers of type c are added at time t. The system relies on existing active couriers (base and previously added on-demand couriers) to handle orders.
- **1**: One courier of type c is added at time t. This courier starts their shift after a 5-minute (300-second) delay at a pickup location with high unassigned order volume.
- **2**: Two couriers of type c are added at time t, both starting after the 300-second delay at the same or different pickup locations, depending on order distribution.

1.3 Objective Function

The objective is to maximize the expected total reward over the operating period under a policy π:

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t \in T_{\text{action}}(\Delta)} r_t(s_t, a_t) \mid s_0\right],$$

where the reward $r_t(s_t, a_t)$ at time t is:

$$r_t(s_t, a_t) = K_{\text{lost}} \cdot n_{t,t+\Delta} + \sum_{c \in \mathcal{C}} K_c \cdot a_t^c.$$

Here, $K_{\text{lost}} = -1$ is the penalty per lost order (not delivered by its due time), $K_{3600} = -0.2$ and $K_{5400} = -0.25$ are costs for adding 1-hour and 1.5-hour couriers, and $n_{t,t+\Delta}$ is the number of orders lost in $[t, t+\Delta)$.

1.4 OPC Policy

In this subsection we answer the question: how the agent decide whether add on-demand courier or not?

The baseline policy for dynamic courier capacity acquisition uses the Orders-Per-Courier (OPC) ratio to decide when to add on-demand couriers at each decision epoch. The policy aims to balance service quality (delivering orders within 2400 seconds of placement) and operational costs, serving as a benchmark for the Deep Q-Learning approach.

The OPC policy calculates the ratio $q_t^{\text{orders}} / \max(q_t^{\text{couriers}}, 1)$, where q_t^{orders} is the number of active orders (pending or assigned but not delivered) and q_t^{couriers} is the number of active couriers at time t. The decision rule is:

- **If** $q_t^{\textbf{orders}} / \max(q_t^{\textbf{couriers}}, 1) > \tau$: Add one 1-hour courier (action $a_t = (1, 0)$ for $c \in \mathcal{C} = \{3600, 5400\}$ seconds), incurring a cost of -0.2. The courier starts after a 300-second delay at a pickup location with high unassigned order volume.
- **If** $q_t^{\textbf{orders}} / \max(q_t^{\textbf{couriers}}, 1) \leq \tau$: Add no couriers (action $a_t = (0, 0)$), relying on existing couriers.

The threshold τ (e.g., 1, 1.5, 2, or 2.5) controls the policy's sensitivity to demand. Only 1-hour couriers are added to ensure responsiveness to short-term demand spikes.

1.5 Constraints

The system operates under the following constraints:

- *Order Assignment*: Orders are assigned greedily, prioritizing those closest to their due times ($m_o = t_o + 2400$ seconds). An order o with pickup location p_o (one of 16 restaurants), dropoff location d_o, and ready time $e_o = t_o + 600$ seconds is assignable to courier q at time $t \geq e_o$ if q can travel to p_o, pick up at $t_{\text{pickup}} = \max\{e_o, t + \text{travel time} + 240\}$, travel to d_o, and deliver by m_o, where 240 seconds is the service time for pickup and dropoff.
- *Courier Availability*: Only couriers active at time t (start time $\leq t <$ end time) can be assigned orders. On-demand couriers start after a 300-second delay.
- *Action Limits*: For each $c \in \mathcal{C}$, $a_t^c \in \{0, 1, 2\}$.
- *Stochastic Demand*: Orders arrive dynamically with random pickup and dropoff locations, following one of four demand patterns with peak periods.
- *Courier Repositioning*: Idle couriers reposition to the nearest restaurant after completing or when unassigned, affecting future assignment feasibility.

2 Summary

The MDP model optimizes courier capacity acquisition by adding on-demand couriers every 300 seconds to maximize expected rewards, balancing the cost of lost orders and courier additions. Spatial and temporal demand uncertainties are addressed through greedy order assignments and courier repositioning, with the state capturing aggregate system dynamics.

References

[1] R. Auad, A. Erera, and M. Savelsbergh. Dynamic courier capacity acquisition in rapid delivery systems: A deep Q-learning approach. *Optimization Online*, 2023.