Fair Courier Assignment and Dynamic Food Pricing via Multi-Agent Reinforcement Learning with Communication

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

October 25, 2024

Keywords: Food delivery; Markov decision process; Multi-agent Reinforcement learning; Agent communication.

1 INTRODUCTION

This paper addresses a gap in food delivery service research, which often focuses on delivery efficiency but overlooks courier fairness. We introduce a hierarchical multi-agent framework to optimize both pricing and courier selection at the upper level using Differentiated Inter-Agent Learning (DIAL, Foerster *et al.* (2016)) for centralized decision-making, and tackle the routing problem at the lower level through decentralized execution methods. Figure 1 illustrates

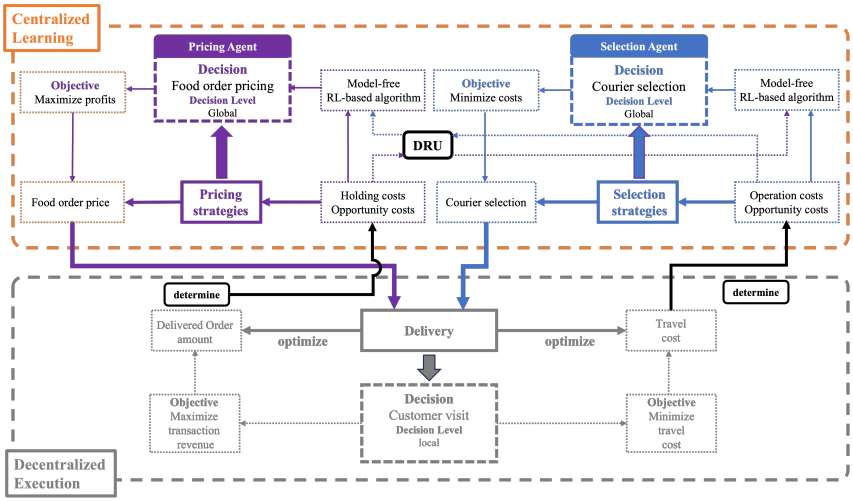


Figure 1 – *The structure of the proposed solution approach*

our multi-agent framework. We model the problem as a Partially Observable Markov Decision Process (POMDP, Foerster *et al.* (2016)). We define two specialized agents at the centralized level: a pricing agent $\mathbf{P}$ to optimize prices, and a courier selection agent $\mathbf{S}$ to assign couriers efficiently. The POMDP is defined as:

$$\left(\mathbf{\Sigma}, \mathcal{S}, \mathbf{J}, \mathcal{Z}, \mathcal{Y}, \hat{\Omega}, \hat{\mathcal{R}}\right), \tag{1}$$

where $\mathbf{\Sigma}$ is the set of centralized agents: the pricing agent $\mathbf{P}$ and the courier selection agent $\mathbf{S}$. $\mathbf{J}$ denotes the set of actions. The pricing agent $\mathbf{P}_\sigma$ makes pricing decisions $a_k(\mathbf{P}_\sigma)$, and the courier selection agent $\mathbf{S}_\sigma$ assigns couriers $a_k(\mathbf{S}_\sigma)$, for $\sigma = 1, 2, \ldots, n$. At each decision stage k, the pricing agent selects actions $a_k(\mathbf{P}_\sigma) \in \mathbf{J}_\mathbf{P}$, and the courier selection agent selects actions $a_k(\mathbf{S}_\sigma) \in \mathbf{J}_\mathbf{S}$. Agents communicate via a protocol, exchanging messages $\xi_k(\mathbf{P}_\sigma), \xi_k(\mathbf{S}_\sigma) \in \Xi_\mathbf{\Sigma}$.

Each agent receives a private observation $z_k(\mathbf{P}_\sigma), z_k(\mathbf{S}_\sigma) \in \mathcal{Z}$, sampled from $z_k(\cdot) \sim \mathcal{Y}(s_k)$. The joint transition function $\hat{\Omega}$ models the system dynamics: $z_{k+1}(\mathbf{\Sigma}) \sim \hat{\Omega}\big(z_k(\mathbf{\Sigma}), a_k(\mathbf{\Sigma})\big)$. The global reward $\hat{\mathcal{R}}$ is shared and decomposed into local rewards $\hat{R}_k(\mathbf{P})$ and $\hat{R}_k(\mathbf{S})$. The objective at stage k_0 is:

$$\max_{a_k(\mathbf{\Sigma})} J\big(z_{k_0}(\mathbf{\Sigma})\big) = \mathbb{E}_{\hat{\Omega}} \sum_{k=k_0}^{K} \gamma^{k-k_0} \hat{\mathcal{R}}_k, \tag{2}$$

with discount factor $\gamma \in (0, 1)$ and horizon K. The local rewards are:

$$\hat{\mathcal{R}}_k = \hat{R}_k(\mathbf{P}) + \hat{R}_k(\mathbf{S}) = \sum_{\sigma=1}^{n} R_k(\mathbf{P}_\sigma) + \sum_{\sigma=1}^{n} R_k(\mathbf{S}_\sigma). \tag{3}$$

After setting pricing and courier assignments, the decentralized agent handles deliveries during stage k. Each stage has $T/\Delta t$ time steps, with simulation interval Δt. The decision variable is $\mathbf{S}_\sigma\big(u_{kh}(t)\big)$, indicating delivery to customer $h \in \mathcal{H}$ at time t. The objective is set as:

$$\max \quad \sum_{t=0}^{T} \sum_{h=0}^{|\mathcal{H}|} F\big(\mathbf{S}_\sigma\big(u_{kh}(t)\big)\big) \tag{4}$$

2 METHODOLOGY

We present solution algorithms for the proposed multi-agent framework by adopting a communication based approach using DIAL.

2.1 Centralized Learning

We decompose the joint action $\mathbf{J}_{\mathbf{\Sigma}}$ into components for each agent, enabling them to learn independent sub-policies mapping observations to their unique action spaces. The optimal action $a_k(\mathbf{P}_\sigma)^*$ satisfies:

$$a_k(\mathbf{P}_\sigma)^* = \arg \max_{a_k(\mathbf{P}_\sigma) \in \mathbf{J}_{\mathbf{P}}} \mathbb{E}_{\hat{\Omega}} \Big[\hat{R}(\mathbf{P}_\sigma) + \gamma J^*_{\mathbf{P}}\big(z_k(\mathbf{P}_\sigma), e_{k-1}(\mathbf{P}_\sigma), \xi_{k-1}(\mathbf{P}_\sigma), a_{k-1}(\mathbf{P}_\sigma)\big) \Big], \tag{5}$$

where $J^*_{\mathbf{P}}(\cdot)$ denotes the optimal cost-to-go given observation $z_k(\mathbf{P})$, hidden state e_{k-1}, communication message ξ_k, and previous action $a_{k-1}(\mathbf{P}_\sigma)$. The optimal value function $J^*_{\mathbf{P}}(\cdot)$ is obtained by iteratively solving the Bellman equation (Bertsekas, 2012):

$$J^*_{\mathbf{P}}(\cdot) = \max_{a_k(\mathbf{P}_\sigma) \in \mathbf{J}_{\mathbf{P}}} \mathbb{E}_{\hat{\Omega}} \Big[\hat{R}(\mathbf{P}_\sigma) + \gamma J^*_{\mathbf{P}}\big(z_k(\mathbf{P}_\sigma), e_{k-1}(\mathbf{P}_\sigma), \xi_{k-1}(\mathbf{P}_\sigma), a_{k-1}(\mathbf{P}_\sigma)\big) \Big]. \tag{6}$$

Similarly, we can obtain the courier selection agent's optimal action $a_k(\mathbf{S}_\sigma)^*$ and $J^*_{\mathbf{S}}(\cdot)$. The optimal value function $J^*_{\mathbf{S}}(\cdot)$ is also able to be obtained by iteratively solving the Bellman equation (Bertsekas, 2012). To further address the 'curse of dimensionality' (Powell, 2011), we use rectified linear units (ReLU, Nair & Hinton (2010)) and gated recurrent units (GRU, Cho (2014)). For simplicity, we denote all agents by $\mathbf{J}_\sigma$ and their embedded observations as $\tilde{z}_k(\mathbf{J}_{\mathbf{\Sigma}})$. The optimal action is then updated as:

$$a^*_k(\mathbf{J}_{\mathbf{\Sigma}}) = \arg \max_{a_k(\mathbf{J}_{\mathbf{\Sigma}})} \mathbb{E}_{\hat{\Omega}} \Big[\hat{\mathcal{R}}_k + \gamma J\big(\tilde{z}_k, \tilde{s}_{k-1}(\mathbf{J}_{\mathbf{\Sigma}}), \tilde{e}_{k-1}(\mathbf{J}_{\mathbf{\Sigma}}), \tilde{\xi}_{k-1}(\mathbf{J}_{\mathbf{\Sigma}}), \tilde{a}_{k-1}(\mathbf{J}_{\mathbf{\Sigma}}); \boldsymbol{\theta}\big) \Big]. \tag{7}$$

Thus, instead of computing exact value functions overall future observations, we estimate the value function $\tilde{J}$ using shared parameters $\boldsymbol{\theta}$ (Sutton & Barto, 2018). This approximation aligns with Deep Recurrent Q-Networks (DRQN, Hausknecht & Stone (2015)), providing stable training

and effective sampling for our discrete action, off-policy, model-free setting. The state embedding rule is defined as (Foerster *et al.*, 2016):

$$\Upsilon_k = \text{TaskMLP}(z_k(\mathbf{J_\Sigma})) + \text{MLP}(\xi_{k-1}(\mathbf{J_\Sigma})) + \text{Lookup}(a_{k-1}(\mathbf{J_\Sigma})) + \text{Lookup}(\mathbf{J_\Sigma}), \tag{8}$$

where MLP integrates various information into a state representation for both environmental and communication actions. TaskMLP transforms raw observations into an embedded representation capturing relevant task features. Υ_k is passed through a 2-layer RNN with GRUs: $\mathbf{v_{1k}} = \text{GRU}[\Upsilon_k, e_{k-1}]$. The final embedding $\mathbf{v_{2k}}$ generates $Q_k(\Sigma), a_k(\Sigma)$ as $Q_k(\Sigma), a_k(\Sigma) = \text{MLP}[\mathbf{v_{2k}}]$. We further use a communication-based recurrent neural network (C-RNN) to approximate the optimal value function $J^*(\cdot)$ as $\tilde{J}(\cdot)$ parameterized by $\boldsymbol{\theta}_k$. The Temporal Difference (TD) error is computed based on the Bellman equation (Sutton & Barto, 2018):

$$\zeta_k = \left[\hat{\mathcal{R}}^*_k + \gamma \tilde{J}(\mathbf{J}^*_{\mathbf{\Sigma}k+1}; \boldsymbol{\theta}'^{(\phi)}) - \tilde{J}(\mathbf{J}_{\mathbf{\Sigma}k}; \boldsymbol{\theta}^{(\phi)})\right]. \tag{9}$$

Here, $\hat{J}(\mathbf{J}_{\mathbf{\Sigma}k}) = \hat{\mathcal{R}}^*_k + \gamma \tilde{J}(\mathbf{J}^*_{\mathbf{\Sigma}k+1}; \boldsymbol{\theta}'^{(\phi)})$ is the target value using target parameters for approximating $\tilde{J}$ at each k. The loss function is defined as $\min_{\boldsymbol{\theta}} \mathcal{L}_{\tilde{J}(\mathbf{J_\Sigma})}(\boldsymbol{\theta}) = \mathbb{E}[\zeta_k^2]$. Consequently, we can employ various reinforcement learning algorithms to address this challenge. Here, we propose the use of Deep Q-Networks (DQN, Mnih *et al.* (2015)) to train centralized agents.

2.2 Decentralized Execution

We address decentralized retail challenges by modeling them as an Orienteering Problem (OP) (Gunawan *et al.*, 2016). Couriers are classified into 'Gold,' 'Silver,' and 'Bronze' tiers to represent different performance levels. Each tier uses a different heuristic approach to solve the Equation (4) for delivery assignments, simulating varying execution performances from different tiers of couriers.

3 Results

Figure 2 illustrates the training performance. The communication-enabled model (DQN-comm) achieves higher and more stable rewards over 100 episodes compared to the non-communication method(DQN-non-comm).

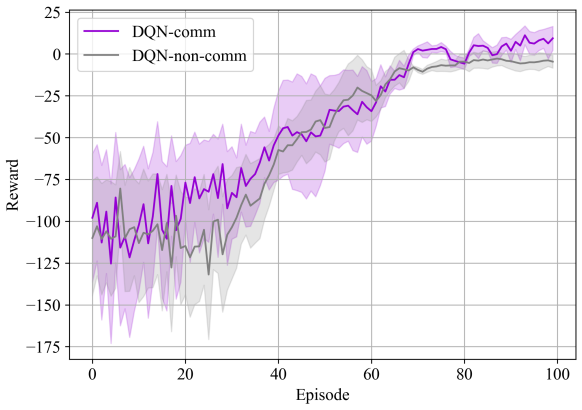
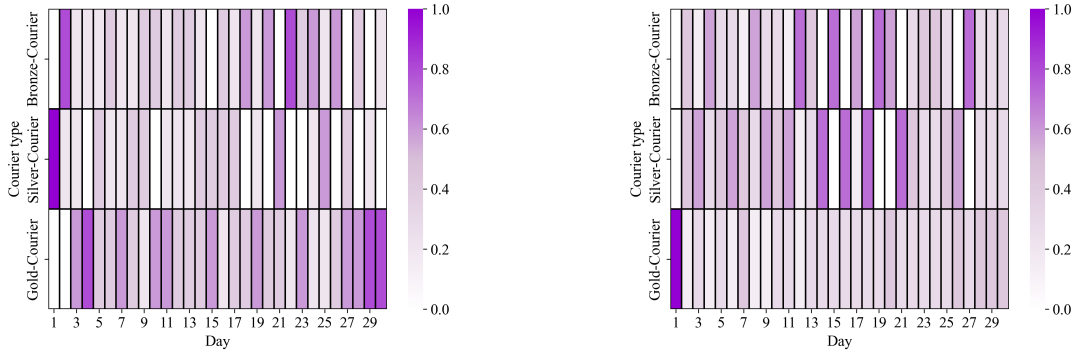


Figure 2 – *Training performance comparison of models with and without communication*

We also test both methods over a 30-day simulation. The simulation uses a 10×10 grid world with 50 customers ordering food according to a known distribution. Figure 3 presents heatmaps comparing courier workload distribution across three tiers: Gold, Silver, and Bronze. These illustrate the advantages of the communication-based system. Without agent communication

(Figure 3a), Bronze and Silver couriers experience unfair workloads. With agent communication (Figure 3b), the workload is more balanced across all tiers, as the system dynamically adjusts courier assignments.



(a) *Courier assignment distribution without agent communication*

(b) *Courier assignment distribution with agent communication*

Figure 3 – *Heatmap comparison of courier assignment distribution with and without communication*

4 DISCUSSION

We introduced a multi-agent control framework that integrates centralized pricing optimization and courier selection with decentralized routing in food delivery services. The objective of the proposed problem is to maximize profitability while ensuring fair job opportunities for couriers in uncertain environments. Our system emphasizes a human-centric approach by providing equitable work opportunities for couriers with different capabilities.

5 REFERENCES

References

Bertsekas, Dimitri P. 2012. *Dynamic Programming and Optimal Control*. 4th edn. Vol. II. Belmont, MA, USA: Athena Scientific.

Cho, Kyunghyun. 2014. On the Properties of Neural Machine Translation: Encoder-decoder Approaches. *arXiv preprint arXiv:1409.1259*.

Foerster, Jakob, Assael, Ioannis Alexandros, De Freitas, Nando, & Whiteson, Shimon. 2016. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, **29**.

Gunawan, Aldy, Lau, Hoong Chuin, & Vansteenwegen, Pieter. 2016. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, **255**(2), 315–332.

Hausknecht, Matthew, & Stone, Peter. 2015. Deep recurrent q-learning for partially observable mdps. *In: 2015 aaai fall symposium series*.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, *et al.* 2015. Human-level control through deep reinforcement learning. *nature*, **518**(7540), 529–533.

Nair, Vinod, & Hinton, Geoffrey E. 2010. Rectified linear units improve restricted boltzmann machines. *Pages 807–814 of: Proceedings of the 27th international conference on machine learning (ICML-10)*.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd edn. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: Wiley.

Sutton, Richard S, & Barto, Andrew G. 2018. *Reinforcement learning: An introduction*. MIT press.