



# An Order Control Policy in Crowdsourced Parcel Pickup and Delivery Service

Yuncheol Kang<sup>(✉)</sup>

Department of Industrial Engineering, Hongik University,  
Wausan-ro 94, Mapo-gu, Seoul, South Korea  
yckang@hongik.ac.kr

**Abstract.** Crowdsourced parcel delivery service has progressed dramatically by actively incorporating innovative technologies and ideas. Yet, maximizing profitability of this new type of delivery service becomes another challenge for service providers as market grows. In this paper we study a service order control policy to maximize profitability from a service provider perspective. Specifically, we suggest an order admission control approach that determines acceptance or rejection of an incoming order according to its profitability characteristics. For this, we model the problem as an average reward Semi-Markov Decision Process and utilize reinforcement learning to obtain an optimal order control policy that maximizes overall profitability of a service provider. Through numerical illustrations, we show that our suggested approach outperforms traditional methods, especially when the order arrival rate is high. Thus, smart order management is an important component of parcel pickup and delivery services.

**Keywords:** Reinforcement learning · Crowdsourced parcel delivery  
Planning and decision-makings · Admission control · Smart logistics

## 1 Introduction

Since emergence of Industry 4.0, a crowdsourced parcel delivery service based on the idea of the sharing economy and just-in-time delivery, a multitude of scholars from different perspectives [1–3] have investigated the utility of this concept. First, crowdsourced parcel delivery is cost efficient in that it uses a shared resource. Second, it can provide more customized services, that is, pickup and delivery at the time a customer specifies. Lastly, it is eco-friendly since fewer vehicles are required to perform the given delivery service, thereby reducing the environmental pollution caused by high volume traffic and oversized delivery vehicles. There are three mainstream transport models that are classified by their type of vehicle: Taxi [4], multi-model, and private vehicle. Among these, the private vehicle is the most flexible and efficient, with the route and ability for pickup and delivery totally dependent on the driver's will. In addition, a private vehicle can support multiple pickup and delivery with less constraints, compared to other transport models. Lee et al. studied a private vehicle routing problem with an integrated decision-making framework that could handle both on-demand parcel delivery and green logistics, such as considering fuel consumption and

gas emissions [5]. In particular, they argue that managing orders in a “smart” way is as important as optimizing operational aspects, such as maximizing profits and minimizing costs occurring in logistics systems. In their study, they assume that the incoming order can be effectively controlled to maximize profitability. Namely, an incoming order that enters a crowdsourced parcel delivery system can be either accepted or declined depending on the profitability associated with that parcel and its delivery. This order control problem maximizes benefits and minimizes costs and is defined and solved through a Markov Decision Process (MDP)-based admission control approach, which decides optimal sequence decisions and whether to include new incoming requests into the appointment schedule [6, 7].

In this paper we focus on optimizing operational costs occurring during conducting on-demand crowdsourced parcel delivery business. Specifically, we use a reinforcement learning approach to effectively manage delivery of orders, thereby maximizing profitability for the delivery business. In the next section, we define a problem that we consider in this paper. Then we suggest a reinforcement learning approach to efficiently solve the problem and illustrate the approach with some numerical examples to reveal the underlying characteristics of the suggested order control algorithm.

## 2 Problem Definition

To begin with, let us consider a single vehicle routing problem (VRP) with soft time windows and multiple on-demand pickup and delivery constraints. Unlike a traditional VRP problem, we assume that information about traveling locations and due-dates are not known to a service provider in advance. In other words, an order can arrive at any time without prior notice. An order contains information about when and where to travel to pickup and deliver a parcel. Once an order arrives, an order control system decides whether to accept the order with consideration for profitability in executing the order. If the order control system accepts the order request, it then schedules pickup and delivery of the parcel into the routing schedule. For the criteria for determining order acceptance, we consider both vehicle information and parcel information. For vehicle information, we consider the current available capacity and the “busyness” of the current schedule. For parcel information, we consider traveling distance and due-date for pickup and delivery as well as the parcel’s load. An order could be rejected due to lack of profitability. If order control rejects the order, it may requote the parcel service. In the following section, we show how this type of problem can be modeled as a sequential decision-making problem, which is an MDP-based model.

## 3 Model Formulation

From an order control perspective, orders keep arriving at random with random quote conditions. With every order arrival, the order control system needs to decide whether to accept or reject the order. From a modeling perspective, this is a typical form of a sequential decision-making problem formulated as a MDP problem. The classical form of MDP, however, is often intractable. To avoid this difficulty, a reinforcement

learning (RL) approach is used to solve the MDP problem, and it is often combined with the function approximation technique, which boosts computational efficiency. In this section we define an MDP model for the problem and RL approach to efficiently solve the model we formulate.

### 3.1 MDP Model for Order Management

As for the ingredients in the MDP problem, we need to define *state*, *action*, and *reward*. First, a vehicle's information and a parcel's information can be regarded as *state*, denoted as  $s \in S := \vartheta \cup \zeta$  where  $\vartheta$  and  $\zeta$  represent a set of vehicle information and parcel information, respectively. A set of vehicle information,  $\vartheta$ , consists of available capacity and "busyness" of the schedule at the current time. A set of parcel information,  $\zeta$ , consists of due-date and traveling distance for pickup and delivery of the parcel and the load of the parcel. In particular, such parcel information is assumed to be unavailable until the corresponding order arrives, which can be regarded as an *event* in our system. Next, an *action*, denoted by  $a \in A$ , takes place in reference to an incoming order, that is, acceptance as is or rejection of the order. The best action is determined by comparing the benefits of acceptance with that of rejection. Such benefit is defined as a *reward* in the MDP setting and denoted by  $r$ , which is governed by  $S$  and  $A$ . With all these elements, we want to obtain an optimal policy that maximizes average reward in executing orders over the long run.

Theoretically, such an MDP model can be formulated as Bellman's equation and solved optimally by using several algorithms, such as value iteration, policy iteration, or linear programming in an infinite horizon of decision settings. Practically, however, the MDP model suffers from the curse of dimensionality, that is, solving the model becomes intractable as the size of problem increases. In our case, the combination of possible state  $\vartheta$  and  $\zeta$  exceeds 10 million, which is impossible to be solved by using a conventional algorithm. Besides, we assume that an order arrives randomly, and its acceptance will be determined at each arrival point in the system. Therefore, inter-decision time is not consistent; thus, the problem needs to be modeled as Semi-MDP (SMDP), which has more complex computational structures. In the following section, we suggest RL with the function approximation approach to solve this model.

### 3.2 Reinforcement Learning (RL) Approach

Q-learning, one of the RL approaches, is an iterative learning algorithm that learns from numerous trials by simulating actions, thus obtaining the best policy for a specific situation. Usually, Bellman's equation can be solved optimally using the Q-learning approach. In this paper the original version of the Q-learning approach is tailored for solving an average reward SMDP problem. Bertsekas proves that a regular Markov chain can be transformed into a Markov chain with an artificial terminate state, and the optimal policy of average reward SMDP for both cases is identical. Such a transformed model is called a stochastic shorted path (SSP) problem and is highly tractable, compared to the original problem [8]. This SSP problem can be solved in a Q-learning form as follows:

$$Q(i, a) = (1 - \alpha)Q(i, a) + \alpha[r(i, a, j) - \rho t(i, a, j) + \mathbb{I}_{j \neq i^*} \max_{v \in A(j)} Q(j, v)] \quad (1)$$

Here  $i, j \in S$  and  $i^* \in S$  is an artificial terminate state.  $\alpha$  stands for a step size for an iterative algorithm.  $t(i, a, j)$  stands for sojourn time from the state  $i$  to state  $j$ , and  $\rho$  is an estimated average reward, which is updated iteratively with accumulated reward and time [9].  $\mathbb{I}_{j \neq i^*}$  is 1 when  $j \neq i^*$ , otherwise 0.

Once an order arrives, its acceptance is determined by comparing the Q value of acceptance with the Q value of rejection. If  $Q(s, a_{accept})$  is greater than  $Q(s, a_{reject})$  then the system accepts the order, otherwise the order is rejected. The difference between the two Q values is defined as the ‘‘additional-fee-for-acceptance.’’ In the case of rejection, the ‘‘additional-fee-for-acceptance’’ becomes a positive value, which means the order would be accepted (i.e., the rejection decision overturned) if the customer were to pay an amount greater than or equal to the ‘‘additional-fee-for-acceptance’’ value. This value is denoted as  $\delta(s)$  and defined as follows:

$$\delta(s) = \begin{cases} Q(s, a_{reject}) - Q(s, a_{accept}), & \text{if } Q(s, a_{reject}) > Q(s, a_{accept}) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Namely, overturning the rejection decision requires  $\delta(s)$  as an additional price. If the incoming order is rejected with higher  $\delta(s)$ , we can argue that the order is less attractive to the service provider. Meanwhile, if the order is rejected with lower  $\delta(s)$ , the rejected decision may be overturned and the order accepted by slightly changing the quote conditions, which makes the order more attractive.

RL can be more efficient in solving models when it incorporates a function approximation scheme. A popular approximation approach is to utilize an Artificial Neural Network (ANN) to efficiently retrieve and store enormous number pairs consisting of state and action. Although the detailed scheme is not included in this paper due to page limits, interest readers are referred to the relevant literature [8–11].

## 4 Numerical Illustration

This section describes the order control algorithm suggested in this paper and presents some numerical illustrations. For convenience, we consider a single vehicle routing problem with soft time windows. Since the order planning and route scheduling part is separated, choosing which route scheduling algorithm is trivial, i.e., no matter which route algorithm is used, the planning trend does not change. In this paper, we use the algorithm which was developed for deriving on-demand just-in-time delivery schedules based on the continuous variable feedback control approach [12].

### 4.1 Experimental Scenarios

In Table 1 we summarize the experimental conditions used for the illustration. All locations and due-date for pickup and delivery are randomly generated under the given condition in Table 1. For representing ‘‘busyness’’ of the schedule, we define the

penalty parameter,  $\zeta$ , as the summation of the deviation between scheduled time and appointment time of all current accepted orders (i.e., earliness or tardiness from the original schedule). Large  $\zeta$  means the current delivery schedule is far from the original appointed due-date. In that case we impose a high penalty for the order, and this will increase the cost for executing the order, thereby lowering the chances of it being accepted. Here, a cost consists of delivery distance and parcel load,  $\zeta$  penalty, and due-date. As (1) the vehicle travels longer, (2) due-date is tighter, (3) the current schedule becomes busier (i.e., high level of deviations between scheduled and appointed time) and (4) the parcel is heavier, we assume the cost for carrying the corresponding order increases. The lower and upper bounds related to cost were set at \$4.99 and \$10, respectively, by referring to the related literature [13].

**Table 1.** Experimental conditions

| Parameter              | Value   |
|------------------------|---|
| Arrival rate           | Varies from 1 to 10 per minute  |
| Vehicle capacity (lbs) | 400   |
| Delivery space         | 20 mile by 20 mile  |
| Due-date               | 0.5, 1, 1.5, 2, 4, 6, 8, and 12 h after current time  |
| $\zeta$ penalty        | No penalty (<0.5 h difference), Light (0.5–1 h difference), Heavy (2> hour difference), Maximum (>3 h difference) |
| Load of a parcel (lbs) | 10, 20, 30, 40  |

## 4.2 Experiment Results and Discussions

With RL and all of the conditions mentioned above, we calculated average reward and the ratio of admission when a new order arrives by varying the arrival rate of the order. The detailed results are shown in Table 2. From the results, we could see the average reward increases gradually as the arrival rate increases. Intuitively, this result is reasonable in that we expect more incoming orders as the arrival rate increases, thereby contributing to increased average reward.

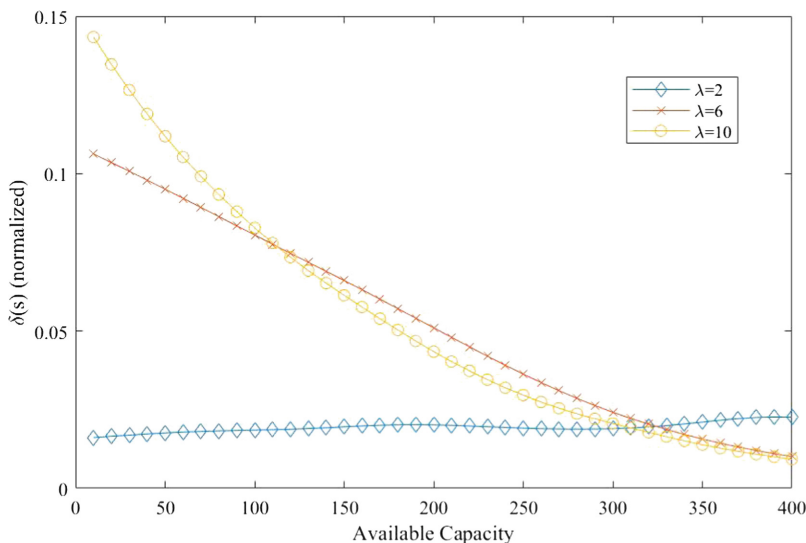
Meanwhile, we could also see the ratio of admission tends to be lower as the arrival rate increases, even though average reward increases at that time. This result can be interpreted as our model tending to choose more profitable orders from among the many incoming orders.

For comparison purposes, we consider a simple heuristic algorithm. The rule of the algorithm is that we accept a parcel as long as vehicle capacity remains. Namely, the algorithm only rejects an order when the capacity becomes full. Intuitively, this type of heuristic makes sense in that the worker can easily apply the rule on site. From a simulation result of the algorithm, however, we saw that average reward decreases as the arrival rate increases. If the system follows this rule, the vehicle’s capacity becomes full as the arrival rate increases. “Full” capacity means increasing fuel costs, thereby lowering profitability to carry the orders.

**Table 2.** Average reward and admitted ratio by arrival rate

| Arrival rate | Average reward (\$) | Admitted ratio |
|--------------|---------------------|----------------|
| 1            | 1.752317895         | 1              |
| 2            | 2.030466464         | 0.605830336    |
| 3            | 1.680257417         | 0.507339358    |
| 4            | 2.120772918         | 0.535525655    |
| 5            | 2.464457988         | 0.208630454    |
| 6            | 3.669486354         | 0.354101483    |
| 7            | 3.453952971         | 0.58208095     |
| 8            | 3.133908961         | 0.541404358    |
| 9            | 4.774441287         | 0.252209083    |
| 10           | 8.397353574         | 0.340053004    |

In terms of “additional-fee-for-acceptance”, we plotted three cases that have different arrival rate values ( $\lambda = 2, 6, \text{ and } 10$ ) in Fig. 1. In the case of a low arrival rate ( $\lambda = 2$ ), we observe  $\delta$  does not significantly change across the range of capacity, meaning that the attractiveness of rejecting the order is trivial in most cases. Intuitively, this result makes sense since we may not need to be “choosy” when an order arrives intermittently. Also, we observe low level of  $\delta$  for the case of low arrival rate, compared to higher arrival cases. Low  $\delta$  means a minimal additional fee will lead to overturning the rejected decision. In other words, even if an order is rejected for some reason(s), it can be relatively easy to have the order accepted by paying a little more.

**Fig. 1.** “Additional-fee-for-acceptance” given available capacity

However, this trend changes when the arrival rate is high ( $\lambda = 6$  and  $10$ ). In this case a high level of  $\delta$  is observed at a lower level of remaining capacity, meaning that a rejected order carries a higher fee for the rejection to be overturned and the order to be accepted. Our algorithm becomes “choosy” in accepting an order when available capacity decreases. This can be interpreted as follows: given the case of highly frequent arrival events, if we have little available capacity, the system does not accept any orders since it expects greater profitability will result from order rejection and re quoting at a higher price for order acceptance. Meanwhile, if there is enough room to load additional parcels, overturning rejected orders is possible by paying a little more.

## 5 Conclusion and Future Research

In this paper we study order control for maximizing profitability for a crowdsourced parcel pickup and delivery service. We show how the MDP-based control approach can be applied to this type of problem. We use a tailored RL approach with an adaptation of function approximation to efficiently solve the SMDP problem, which is often intractable in determining optimal policy. From the numerical results we gained some insights on how order control policy behaves as a variety of parameters such as arrival rate or available capacity changes.

Although we consider only two actions for this problem, that is, *accept* or *reject* the parcel, we can further add other types of actions. In particular, in the case of rejecting an order, we regard this as the starting point for negotiating with a customer. In other words, the order control system may suggest that the customer resubmit the order with at least one of the following conditions: (1) select different time slot with greater availability or (2) pay higher price for order. Thus, we may need to guide the customer by presenting different quote conditions until the request can finally be accepted. In addition, this paper only deals with a single vehicle routing problem but can be extended to a multiple vehicle routing problem [14], which represents the future direction of this research.

**Acknowledgements.** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1C1B1005354).

## References

1. Wang, G., Gunasekaran, A., Ngai, E.W., Papadopoulos, T.: Big data analytics in logistics and supply chain management: certain investigations for research and applications. *Int. J. Prod. Econ.* **176**, 98–110 (2016)
2. Davenport, T.H., Glaser, J.: Just-in-time delivery comes to knowledge management. *Harv. Bus. Rev.* **80**, 107–111 (2002)
3. Lai, K., Cheng, T.E.: *Just-In-Time Logistics*. Routledge, Abingdon (2016)
4. Li, B., Krushinsky, D., Reijers, H.A., Van Woensel, T.: The share-a-ride problem: People and parcels sharing taxis. *Eur. J. Oper. Res.* **238**, 31–40 (2014)
5. Lee, S., Kang, Y., Prabhu, V.V.: Smart logistics: distributed control of green crowdsourced parcel services. *Int. J. Prod. Res.* **54**, 6956–6968 (2016)

6. Wu, C.-H., Chien, W.-C., Chuang, Y.-T., Cheng, Y.-C.: Multiple product admission control in semiconductor manufacturing systems with process queue time (PQT) constraints. *Comput. Ind. Eng.* **99**, 347–363 (2016)
7. Kim, C., Dudin, S.: Priority tandem queueing model with admission control. *Comput. Ind. Eng.* **61**, 131–140 (2011)
8. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-dynamic programming: an overview. In: Proceedings of the 34th IEEE Conference on Decision and Control, pp. 560–564. IEEE (1995)
9. Gosavi, A.: Simulation-based optimization: an overview. In: Gosavi, A. (ed.) *Simulation-Based Optimization*. ORSIS, vol. 55, pp. 29–35. Springer, Boston, MA (2015). [https://doi.org/10.1007/978-1-4899-7491-4\\_3](https://doi.org/10.1007/978-1-4899-7491-4_3)
10. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*, 2nd edn. MIT Press, Cambridge (2017)
11. Bertsekas, D.P.: *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont (1995)
12. Lee, S., Prabhu, V.V.: Just-in-time delivery for green fleets: a feedback control approach. *Transp. Res. Part Transp. Environ.* **46**, 229–245 (2016)
13. Rougès, J.-F., Montreuil, B.: Crowdsourcing delivery: new interconnected business models to reinvent delivery. In: 1st International Physical Internet Conference, pp. 1–19 (2014)
14. Kafle, N., Zou, B., Lin, J.: Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. *Transp. Res. Part B Methodol.* **99**, 62–82 (2017)