A Deep Reinforcement Learning Approach for Online Parcel Assignment

Hao Zeng
Cainiao Network
Hangzhou, China
zenghao.zeng@cainiao.com

Qiong Wu
Cainiao Network
Hangzhou, China
melody.wq@cainiao.com

Kunpeng Han
Cainiao Network
Hangzhou, China
kunpeng.hkp@cainiao.com

Junying He
Cainiao Network
Hangzhou, China
junying.hjy@cainiao.com

Haoyuan Hu
Cainiao Network
Hangzhou, China
haoyuan.huhy@cainiao.com

ABSTRACT

In this paper, we investigate the online parcel assignment (OPA) problem, in which each stochastically generated parcel order needs to be assigned to a candidate route for delivery with the objective to minimize the total delivery cost under certain business constraints. The OPA problem is challenging due to its stochastic nature: each parcel's candidate routes, which depend on the parcel's attributes, are unknown until its order is placed, and the total parcel volume to be assigned is uncertain in advance. To tackle this problem, we propose an algorithm based on deep reinforcement learning, namely *PPO-OPA*, that shows competitive performance. More specifically, we introduce a novel Markov Decision Process (MDP) to model the decision-making process in the OPA problem, and develop a policy gradient algorithm that adopts attention networks for policy evaluation. By designing a dedicated reward function, our proposed algorithm can achieve a lower total cost with a smaller violation of constraints, compared to the traditional method used in the industry that assigns parcels to candidate routes proportionally. In addition, the performances of our proposed algorithm and the Primal-Dual algorithm are comparable, while the later assumes a known total parcel volume in advance, which is unrealistic in practice.

KEYWORDS

Online Assignment; Reinforcement Learning; Markov Decision Process

ACM Reference Format:
Hao Zeng, Qiong Wu, Kunpeng Han, Junying He, and Haoyuan Hu. 2023. A Deep Reinforcement Learning Approach for Online Parcel Assignment. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 8 pages.

1 INTRODUCTION

The online parcel assignment (OPA) problem naturally arises from today's e-commerce environment, where a logistics company needs to assign each parcel to a candidate route for delivery after customers make online purchases. As shown in Figure 1, a candidate route consists of multiple logistics service providers and physical nodes such as hubs. When a parcel is assigned to a candidate route, it consumes resource of all providers and hubs within the route, and raises a delivery cost to be paid by the logistics company. The set of available candidate routes and their corresponding delivery costs are determined by the parcel's attributes such as origin, destination, weight, parcel type, etc., which remain unknown until the parcel order is made. As online shopping prevails and daily parcel volume grows tremendously, it becomes crucial for the logistics company to make parcel assignments wisely because it could save hundreds of thousands of dollars per day on total delivery cost. Other than delivery cost, business constraints due to resource capacities or established contracts, need to be considered in this problem. A business constraint can be interpreted as the lower and upper bounds of the number of parcels that can be assigned to a provider or hub. The OPA problem is to assign each stochastically generated parcel to a candidate route with the objective as minimizing the total delivery cost subject to given business constraints.

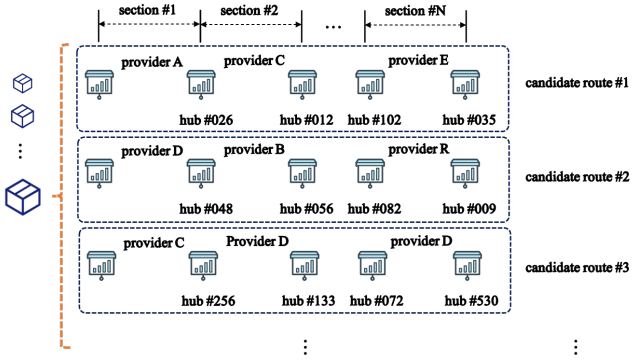

Figure 1: Incoming parcels and their candidate routes in online parcel assignment.

The OPA problem is closely related to several problems that have been studied in the literature. By assuming all incoming parcels' attributes and candidate routes are known, the offline version of the parcel assignment problem can be formulated as a deterministic integer programming problem. If we assume the total parcel volume to be assigned is given, which is not true in real practice, our problem would be similar to the online allocation problem [6, 31]. In the setting of the online allocation problem, the total

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

number of requests is assumed to be given, but the arrival sequence of requests is unknown. Such problem appears in many practices such as adwords matching [11, 18], online routing [7] and online combinatorial auction [9]. For the online allocation problem, there exists a competitive ratio of $(1 - 1/e)$ with adversarial arrivals and $(1 - \varepsilon)$ with stochastic arrivals [8]. Dropping out of the assumption of known total number of requests, the OPA online problem can be viewed as a special kind of the online resource allocation problem under horizon uncertainty [5], where the horizon tends to be large ($> 10^5$ parcels to be assigned per day) but remains unknown until the end of the decision-making process. Besides, the attributes of future parcels (including origin, destination, and weight etc) is unknown and irrelevant to assignment policy, which violates the Markovian property. Therefore, there is no existing problem formulation that is completely applicable to the OPA problem.

Several methods have been attempted for solving the OPA problem. For example, the greedy method, which always assigns the parcel to the route with the lowest cost, can guarantee to minimize the total cost. However, since this method does not take any constraint into account, the possibly severe violation of constraints makes it inappropriate for real practice. Other typical algorithms based on online primal-dual framework [8] has been used to solve a variety of online optimization problem, such as online adwords problem [11], online task assignment in crowd-sourcing markets [15]. Nevertheless, the online primal-dual algorithm requires the total number of parcels given, which is impractical for actual scenarios. Recently, deep reinforcement learning (DRL) approaches have received great attention for their capacity to solve complex decision-making problems efficiently. In this paper, we focus on solving the OPA problem utilizing a specially designed DRL method. More specifically, by using a modified MDP to formulate the OPA problem, we propose a proximal policy optimization (PPO) algorithm, in which assignment decisions are made based on current observation and past information, to optimize the objective while keeping the violation of constraints as small as possible. Through experiments, we show that our DRL approach can be powerful for solving the OPA problem.

The main contributions of this paper are summarized as follows:

- We propose Online Assignment MDP for modeling the decision-making process in the OPA problem, which can also be applied to a variety of online assignment problems.
- Based on the PPO framework, we propose a DRL method that uses attention networks to learn the feature combination of incoming parcel's information and constraints' status for improving the assignment policy.
- In the experiments, we test proposed PPO-OPA algorithm in real datasets from Cainiao Network. The results show that our approach outperforms the traditional assignment method used in the logistics industry. In addition, we show that the performances of our proposed algorithm and the Primal-Dual algorithm are comparable.

2 RELATED WORK

Our algorithm and analysis build on the Markov Decision Process (MDP) framework, which provides a widely applicable mathematical formulation for sequential decision-making problems. In the MDP framework, the agent observes a state s_t from the environment at each time step t, and then makes an action a_t according to its policy $\pi(s_t)$. After an action is taken, the state transits to the next state s_{t+1} and a reward r_t is sent back from the environment to the agent. The goal of such process is to maximize the accumulated discounted reward $R = \sum_{t=1}^{T} \gamma^{t-1} r_t$ where $\gamma \in (0, 1]$. One general algorithm for solving MDP problems is reinforcement learning [25], in which a well-trained agent can learn an optimal policy π to maximize the accumulated discounted reward R through past experience knowledge. Recently, deep reinforcement learning (DRL) methods that employ neural networks for function approximations [19] to handle high-dimensional state and action space show promising results in solving MDP problems. The most successful implementations include AlphaGo [24] and AlphaZero [23], which convincingly defeated the world champion programs in chess, Go and Shogi without any domain knowledge other than underlying rules as inputs during training.

There have been an increasing number of studies on employing DRL methods for industrial decision-making problems. Zhang and Dietterich [32] utilizes the temporal difference learning to learn a heuristic evaluation function over states to develop domain-specific heuristics for job-shop scheduling. Tesauro et al. [26, 27] shows the feasibility of online RL to learn resource valuation estimates that can be used to make high-quality server allocation decisions in multi-application prototype data center scenarios. To minimize power consumption while meeting demands of wireless users over a long operational period, Xu et al. [29] presents a novel DRL-based framework for power-efficient resource allocation in cloud RANs. Du et al. [12] learns a policy that maximizes the profit of the cloud provider through trial and error, where they integrate long short-term memory (LSTM) neural networks into improved DDPG to deal with online user arrivals, to address both resource allocation and pricing problems. Recently, Ye Li and Juang [30] develops a novel decentralized resource allocation mechanism for vehicle-to-vehicle (V2V) communications based on DRL. In summary, most of these studies assume that the decision process is Markovian, and therefore, their methods can not be directly used to solve the OPA problem.

Most useful DRL methods can be categorized into two classes: value learning and policy gradient. Value learning are aimed at explicit learning of value functions from which the optimal policy can be obtained. A commonly used branch of value learning includes Deep Q-Network (DQN) [19] and its variants (e.g., Rainbow [14]). These methods are mainly suitable for discrete action spaces and are successful in mastering a range of Atari 2600 games. The policy gradient methods, on the other hand, attempt to learn optimal policies directly. The policy gradient methods with the assistance of baselines (e.g., value functions) are also referred to as Actor-Critic methods, which are suitable for both discrete and continuous action space. Some representative Actor-Critic methods are DDPG [17], TRPO [21] and PPO [22]. TRPO develops a series of approximations and the original objective of policy gradient is converted to minimization of a surrogate loss function under constraint on the KL divergence between old and new policies, and uses the trust-region method to guarantee policy improvement with non-trivial step sizes. PPO [22] is a substitute of TRPO that is

more applicable to large-scale decision problems. The algorithms mentioned above rely on a common assumption that the problem can be modeled as an MDP problem, which makes them not applicable to some online decision-making problems. To bridge between reinforcement learning and online learning, Even-Dar et al. [13] proposes an online MDP that relaxes the Markovian assumption of the MDP by setting the reward function to be time dependent. Similarly, our work extends the traditional MDP to adapt the OPA problem by introducing an uncertain observation at each step. This method can efficiently enjoy the exploration-exploitation benefits of RL algorithms to acquire an effective policy.

Another class of algorithms that builds on the primal-dual framework has also been applied in a variety of online optimization problems [4, 8]. For example, in the online adwords problem with stochastic assumption that keywords arrive in an online manner, advertisers must be assigned to keywords so that the revenue is maximized without exceeding any advertiser's budget. The online primal-dual algorithm can achieve a near-optimal performance under the case where the total budget is sufficiently large [3]. In addition, Ho and Vaughan [15] introduces the online task assignment problem in crowd sourcing markets, in which workers arrive one by one, and must be assigned to a task. By designing the Dual Task Assigner (DTA) based on the primal-dual framework, this work proves that DTA outperforms other algorithms empirically. However, the DTA algorithm requires the total number of workers to be given in advance. Recently, Balseiro et al. [5] combines dual descent with a carefully-chosen target consumption sequence to solve online resource allocation under horizon uncertainty, which do not require knowing the number of requests, and proves that it achieves a bounded competitive ratio when the horizon grows large.

3 PROBLEM FORMULATION

In this section, we formulate the OPA problem explicitly. For a period of time, we define the total parcel volume as m. In the OPA problem, the decision-making agent needs to assign each incoming parcel to one of its available candidate routes in order to minimize the total delivery cost subject to given business constraints. We use $\mathcal{J}(i)$ to define the set of all candidate routes of the parcel i. Let $\mathcal{K}$ be the set of all constraints. $C(i,k)$ denotes the set of routes corresponding to parcel i and constraint k. In addition, we define the binary decision variables $x_{i,j}$ that is 1 if parcel i is assigned to route j and 0 otherwise. The corresponding cost of parcel i assigned to route j is denoted as $c_{i,j}$. Therefore, the offline version of the parcel assignment problem can be formulated to the following linear programming:

$$\begin{aligned} \min_{x} & \sum_{i=0}^{m} \sum_{j \in \mathcal{J}(i)} c_{ij} x_{ij} \\ \text{s.t.} & \sum_{j \in \mathcal{J}(i)} x_{ij} = 1, \quad \forall i \\ & L_k \le \sum_{j \in C(i,k)} \sum_{i=0}^{m} x_{ij} \le U_k, \quad \forall k \in \mathcal{K} \\ & x_{ij} \in \{0,1\}, \quad \forall j \in \mathcal{J}(i), \forall i \end{aligned} \tag{1}$$

In this paper, we consider two types of business constraints:

- **Capacity Constraints:** the upper and lower bounds of parcel volume that can be assigned to a provider's hub.
- **Proportion Constraints:** for each pair of origin and destination, the upper and lower bounds of percentage of parcels assigned to a provider.

For Capacity Constraints, L_k is usually 0 and U_k is the capacity of hub k. For Proportion Constraints, $L_k = p_k^L \cdot n_k$ and $U_k = p_k^U \cdot n_k$, where p_k^L and p_k^U is given by the business team and n_k denote the number of parcels corresponding to constraint k. It should be noted that n_k is known under the offline setting but unknown in the online setting due to uncertain upcoming parcels.

In the online setting, the parcel assignment problem has an unknown total parcel volume m and unpredictable future parcel information, which is non-Markovian. The agent needs to make an assignment decision for any incoming parcel based on the currently observed parcel's attributes and constraints' states. However, only the constraints' states satisfy the Markov property. One should note that the decision-making process in the OPA problem is different from Partially Observable MDP whose observation depends on the new state or action. Therefore, it is necessary to modify the original MDP for reinforcement learning algorithms to be used for the OPA problem. Here, we present the following definition of Online Assignment MDP that introduces observations $\mathcal{O}$:

Definition 1 (Online Assignment MDP). *An online assignment MDP is a 5-tuple $(\mathcal{O}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where*

- *$\mathcal{O}$ is the set of observations from an unknown distribution,*
- *$\mathcal{S}$ is the set of states,*
- *$\mathcal{A}$ is the set of actions,*
- *$\mathcal{P}$ is the state transition probability defined as*

$$\mathcal{P}^t_{o,s,s'} = \Pr(S_{t+1} = s' | O_t = o, S_t = s, A_t = a),$$

- *$\mathcal{R}$ is the reward function defined as*

$$\mathcal{R}^a_{o,s} = \mathbf{E}(R_{t+1} | O_t = o, S_t = s, A_t = a).$$

It is worth to note that Definition 1 can also be applied to model a variety of online allocation problems with unknown upcoming requests. For example, O_t can denote the online arrival keyword at time t for the online adwords problem, and online arrival worker at time t for the online task assignment problem. We now formulate the specific components for the OPA problem at time t:

- **Observation**: incoming parcel information o_t,
- **State**: Current constraint status s_t. For capacity constraints, $s_t = \{h_i(t)/h_i, i \in \mathcal{H}\}$, where $\mathcal{H}$ is the set of all hubs and h_i is the upper bound of capacity for hub i. For proportion state, $s_t = \{p_j(t), j \in \mathcal{R}\}$, where $\mathcal{R}$ is the set of all routes and $p_j(t)$ is the current ratio for the providers in route j.
- **Action**: action sample from a discrete distribution corresponding each candidate routes of parcel o_t.
- **Reward**: The Design of reward is the most challenging part of the problem. At each time step, the immediate reward should integrate both constraints' states and parcel information. Since the objective is to minimize the total cost, the first part of the reward is the negative of cost of assigning to route c_{a_t}, which depends on action a_t. For a capacity constraint, a smaller the remaining capacity leads to a greater

penalty. For a proportion constraint, a proportion further below the lower bound (or above the upper bound) should have a greater penalty. Hence, the reward is designed as follows:

$$r_t = -c_{a_t} + \lambda f_{a_t}(t), \tag{2}$$

where λ is a hyperparameter to leverage the importance of constraint state function $f_{a_t}(t)$ and cost c_{a_t}. For action a_t corresponding to i-th capacity constraint, the penalty can be written as

$$f_i(t) = e^{-h_i(t)/h_i}; \tag{3}$$

If a_t corresponds to proportion constraint i, then

$$\begin{aligned} f_i(t) &= I(p_i(t) < L_i)(p_i(t) - L_i) \\ &+ I(p_i(t) > U_i)(U_i - p_i(t)), \end{aligned} \tag{4}$$

where $I(\cdot)$ is the indicator function.

Another reward design is to use the negative of cost as reward directly and use the constrained MDP (CMDP) method [1, 10]. In the experiments, we combine Lagrangian relaxation with our proposed DRL algorithm to control constraint violation. However, it can not achieve a better performance compared to methods that add a penalty to reward for online parcel assignment problems.

4 PPO-OPA ALGORITHM

In this section, we present the DRL method based on Proximal policy optimization (PPO) [22] for solving the OPA problem. As mentioned before, PPO is a commonly used RL algorithm with excellent performance for solving a variety of MDP problems. As an Actor-Critic algorithm, the policy function and state value function (often represented by actor network and critic network) need to be estimated during training. PPO adopts the advantage function to assist update gradient and reduce variance. One commonly useful advantage function A_t is based on the temporal-difference (TD) error estimation:

$$A_t = r_t + \gamma V(s_{t+1}) - V(s_t).$$

However, for the OPA problem, the value function also depends on the incoming parcel information. In other words, $V(o_t, s_t)$ is a function on both o_t and s_t. Since o_t is non-Markovian and difficult to predict, so does $V(o_t, s_t)$. Therefore, replacing to estimate future accumulated reward (by critic network), we use reward network $R_\phi(s_t, o_t)$ as the estimator of reward based on the current state and parcel information. Then, we define the advantage A_t as follows:

$$A_t = r_t - R_\phi(s_t, o_t), \tag{5}$$

where $R_\phi(\cdot, \cdot)$ is the reward function which is represented by the reward network. The loss function for this reward network update is

$$E_\phi\left[(R_\phi(s_t, o_t) - r_t)^2\right]. \tag{6}$$

Parcel information includes the parcel's attributes and candidate routes information, which are represented by parcel features and candidate-route features, respectively. One parcel corresponds to multiple candidate routes and different parcels may have different numbers of candidate routes. Therefore, we propose to use an actor network to capture the parcel features and candidate-route features as shown in Figure 2. Parcel features and candidate-route features are inputted to different embedding layers followed by separated

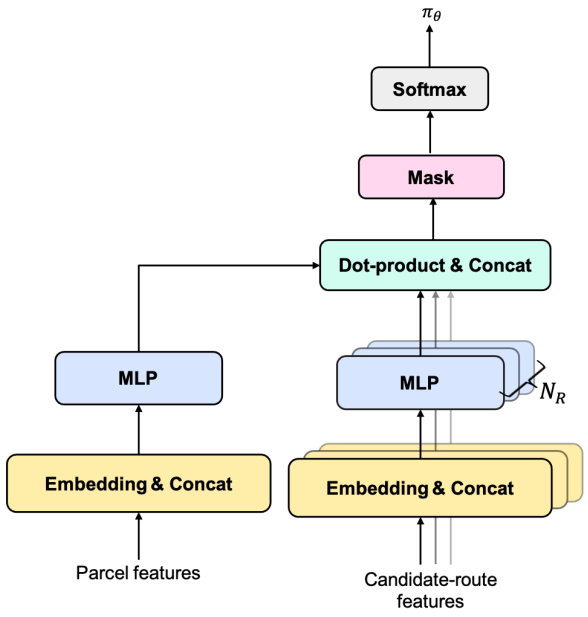

Figure 2: The actor network. Parameter sharing is applied to route vectors in candidate-route features and probabilities of assigning to each route are output from the softmax layer.

Multiple Layer Perceptrons (MLP). For the candidate-route features, we use identical parameters for each route. We define N_R as the maximum number of possible candidate routes per parcel. If the number of candidate routes is less than N_R we will construct fictitious routes with default costs and default constraint states, of which all the values are set to be 0. Then, a mask matrix is used to convert the output of fictitious routes to 0.

The reward network in Figure 3 is similar to the actor network. Parameter sharing is also utilized to accommodate the candidate-route features. Besides, the attention mechanism [28] is employed to calculate the state value for each incoming parcel. An attention function can be described as a mapping from a query and a set of key-value pairs to an output. Here, we treat the parcel features as a query, and candidate-route features are used to generate the key-value pair:

$$q = W^q o, k_i = W^k h_i, \text{ and } v_i = W^v h_i, \tag{7}$$

where $i \in [1, 2, ..., N_R]$. Then, the output is computed as a weighted sum v:

$$v = \sum_{i=1}^{n} q^T k_i v_i. \tag{8}$$

Finally, the reward value can be obtained by

$$R_\phi(s_t, o_t) = \text{MLP}(v). \tag{9}$$

Our network design is simple enough to guarantee fast inference in industrial applications. The experimental results show the effectiveness of this network structure.

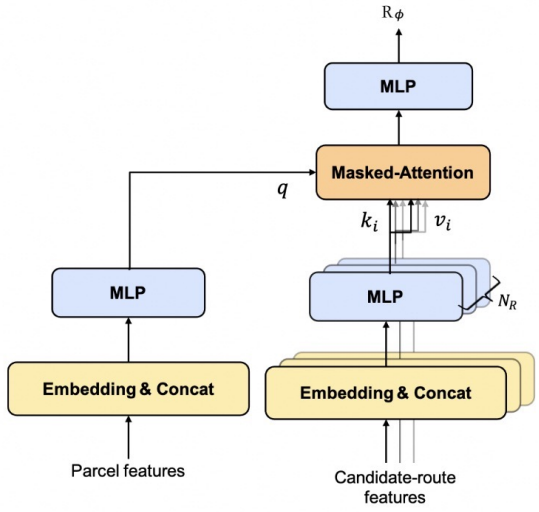

Figure 3: The reward network. Parameter sharing is applied to route vectors in candidate-route features. Masked-Attention layers are used for calculating state value function given a certain parcel and current constraint state.

Based on the PPO algorithm, the improved clipped optimization objective for policy updating is

$$L^{\text{CLIP}}(\theta;\hat{\pi}) = E_{\tau\sim\hat{\pi}}\left[\sum_{t=0}^{T}\min(p_t(\theta;\hat{\pi}), \text{CLIP}(p_t(\theta;\hat{\pi}), 1-\varepsilon, 1+\varepsilon))A_t\right], \tag{10}$$

where $p_t(\theta;\hat{\pi}) = \frac{\pi_\theta(a_t|s_t,o_t)}{\hat{\pi}(a_t|s_t,o_t)}$. Our DRL algorithm is described in Algorithm 1. The trajectories are collected in parallel through policy π_{θ_k} (line 2). Then, the network parameters θ and ϕ are updated by using Adam [16].

5 PERFORMANCE EVALUATION

We implement and evaluate the PPO-OPA algorithm on a workstation computer (ubuntu 16.04), which has an Intel Xeon Platinum 8163 @ 2.50 GHz, 32 GB memory and an Nvidia Tesla V100 GPU with 16 GB memory. We use PyTorch [20] for implementation. For the neural network setting, we set the embedding dimensions as 64 in both the actor and reward networks. For the actor network, the MLP part for parcel features has a single layer of 128 neurons, while that for candidate-route features has two layers: one has 256 neurons and the other has 128 neurons. All layers are with ReLu activation. For the reward network, the MLP parts have the same settings as those in the actor network. After the masked-attention layer, the last MLP part has a layer with 64 neurons with Sigmoid activation followed by a linear layer with 1 neuron. The learning rates for the actor and reward networks are both set to be 10^{-3}. The importance hyperparameters λ's are set to be 10 and 300 for the capacity and proportion constraints, respectively.

Datasets: we use two datasets, denoted as dataset #1 and dataset #2, both of which are real data provided by Cainiao Network. Each

Algorithm 1 Proximal policy optimization for online parcel assignment (PPO-OPA)

Input: initial policy parameters θ_0 and initial value function parameters ϕ_0.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Collect set of trajectories $D_k = \{\tau_i\}$ by running policy π_{θ_k} in the environment.
3: Compute rewards r_t for each trajectory.
4: Compute advantage estimates, $A_t = r_t - R_{\phi_k}(s_t, o_t)$.
5: Update policy by maximizing the PPO objective:

$$\pi_{k+1} = \arg\max_\theta \frac{1}{|D_k|T}\sum_{\tau\in D_k} L^{\text{CLIP}}(\theta, \pi_{\theta_k}),$$

typically via stochastic gradient descent with Adam.
6: Fit reward function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_\phi \frac{1}{|D_k|T}\sum_{\tau\in D_k}\sum_{t=0}^{T}\left(R_\phi(s_t, o_t) - r_t\right)^2,$$

typically via stochastic gradient descent with Adam.
7: **end for**

dataset contains two parts of data, namely parcel data and constraints configuration data.

- Parcel data: This data contains the records of historical parcels created within a country and a time period, sorted by their creation times. Each record shows one parcel's information, including its attributes, candidate routes and corresponding costs.
- Constraints configuration data. This data contains the configuration of business constraints, such as capacity constraints and proportion constraints, that should be considered while making the assignment decisions.

Dataset #1 contains 625 hub capacity constraints and the daily parcel volume varies from 567429 to 806824. On the other hand, dataset #2 contains only 51 proportion constraints and the daily parcel volume is smaller in general, ranging from 293208 to 326332.

In the training procedure, we select the parcel data of datasets #1 and #2 created within a particular day T. That is, 684793 records from dataset #1 and 308329 from dataset #2 are selected. The agent uses this data for trajectory collection and trains neural networks about 20 episodes for attaining convergence. In each episodes, we first collect 50 trajectories in parallel and put all MDP tuples in the trajectories into a buffer. Then, we shuffle the buffer and update the parameters of the actor and reward networks using Adam. The mini-batch size for gradient descent is 2048. In the validation procedure, we use the parcel data of datasets #1 and #2 created within the next three days (i.e., $T+1$, $T+2$, $T+3$).

We compare the results from PPO-OPA against those from three other online algorithms and the integer programming (IP) method, descriptions of which are as follows:

(1) **IP**: the OPA problem can be formulated as an IP problem (1), if all parcels are known in advance. It is straightforward that the solution to (1) is optimal for the OPA problem. Therefore, we can use the IP gap, the difference between the optimal

Dataset #1	Algorithm	Average Cost	IP Gap	Violation Rate
$T+1$	PPO-OPA	100.73	0.0688%	2.53%
	PPO-PD	102.05	1.3834%	4.20%
	Proportion	101.05	0.3874%	2.50%
	PDO	100.67	0.0671%	2.50%
	IP(offline)	100.66		
$T+2$	PPO-OPA	99.782	0.0662%	6.32%
	PPO-PD	101.24	1.5242%	8.95%
	Proportion	100.19	0.4723%	6.28%
	PDO	99.719	0.0030%	6.26%
	IP(offline)	99.716		
$T+3$	PPO-OPA	98.479	0.0872%	5.44%
	PPO-PD	100.47	2.1148%	9.33%
	Proportion	98.927	0.5457%	5.37%
	PDO	98.390	-0.0027%	5.39%
	IP(offline)	98.393		

Table 1: The evaluation results in dataset #1 with capacity constraints. The agent is trained by using the parcel data from day T. The parcel volumes for $T+1$, $T+2$, $T+3$ are 567429, 756579 and 806824 respectively.

Dataset #2	Algorithm	Average Cost	IP Gap	Violation Rate
$T+1$	PPO-OPA	81.193	-0.1276%	2.57%
	PPO-PD	81.130	-0.2052%	5.54%
	Proportion	81.459	0.1993%	3.39%
	PDO	81.139	-0.1946%	5.37%
	IP(offline)	81.297		
$T+2$	PPO-OPA	78.565	0.0495%	3.31%
	PPO-PD	78.472	-0.0685%	8.00%
	Proportion	78.723	0.2509%	4.87%
	PDO	78.493	-0.0419%	2.42%
	IP(offline)	78,526		
$T+3$	PPO-OPA	84.753	0.1213%	2.25%
	PPO-PD	84.659	0.0105%	6.95%
	Proportion	84.930	0.3308%	3.31%
	PDO	84.683	0.0392%	2.06%
	IP(offline)	84.650		

Table 2: The evaluation results for dataset #2 with proportion constraints. The agent is trained by using the parcel data from day T. The parcel volumes for $T+1$, $T+2$, $T+3$ are 293208, 322391 and 326332 respectively.

objective value and the objective value from certain algorithm, as a measurement of performance. To solve (1), we use SCIP [2], a commonly used solver for IP problems.

(2) **Proportion**: this is a traditional method used for online parcel assignments. The proportion algorithm relies on the IP solutions from historical parcel data. Here, we collect 30 days' parcel data (total parcel volume > 10 million) before and on day T, and solve the offline IP problems. Then, we summarize all the assignments and compute the proportion of parcels assigned to each candidate route. For any incoming parcel, the algorithm randomly assign it to one of the candidate routes with probabilities in proportion to the above proportions computed beforehand.

(3) **PDO**: the primal dual optimization (PDO) [8] is a powerful technique for a wide variety of online problems. In this experiment, we run the PDO algorithm for solving (1), where Lagrangian relaxation is used to control constraint and dual variables would be updated at each iteration. For this algorithm, we use the actual total daily parcel volume as input, which is impossible to acquire in real practice.

(4) **PPO-PD**: For reinforcement learning, Lagrangian relaxation is an effective technique to process soft constraints. Here, we set the reward to be the negative of cost in the Online Assignment MDP framework and use the primal-dual update to control the violation of constraints, which is similar to Chow et al. [10] and leads to the unconstrained problem,

$$\min_{\lambda \geq 0} \max_{\theta} L^{\text{CLIP}}(\theta; \pi) - \sum_{k \in \mathcal{K}} \lambda_k (J_k(\pi) - U_k),$$

where $J_k(\pi)$ represents the capacity from policy π for constraint k.

Accordingly, the performance metrics are:

- Average cost: the total cost of assigned parcels divided by the number of assigned parcels.
- IP gap: the difference between the average cost of the IP solution and the average cost of the compared algorithm's solution, divided by the average cost of the IP solution.
- Violation rate: the number of parcels that violate constraints divided by the total number of parcels. IP solution has zero constraint violation rate for hub capacity constraints and route proportion constraints since IP solution is the optimal solution solved in an offline manner.

Table 1 and 2 show the average cost of parcels, IP gap and violation rate achieved by PPO-OPA, PPO-PD, Proportion, PDO and IP using dataset #1 and dataset #2. PPO-OPA achieves about 0.2-0.3% cost reduction and fewer constraint violation rates than the proportion and PPO-PD algorithms. Moreover, PPO-OPA trained by one-day data has almost the same performance as PDO with known parcel volume. From this result, we claim that our method is more suitable for real scenarios, because it does not require a known daily parcel volume but still can achieve a competitive performance.

6 CONCLUSION

In this paper, we introduce the online parcel assignment (OPA) problem, which is aimed at assigning each incoming parcel to a candidate route for delivery in order to minimize the total cost under consideration of given business constraints. Several challenges exist in this problem, including the large but uncertain number (beyond 10^5) of daily parcels to be assigned, the variability of parcels' attributes and the non-Markovian characteristics of parcel arrival dynamics. To tackle this problem, we propose the Online Assignment MDP and present a DRL approach named PPO-OPA. In this approach, Proximal Policy Optimization (PPO) is adopted with a specifically designed MDP for conducting the online assignment. The actor and reward networks adopt the attention mechanism and parameter sharing to accommodate each incoming parcel with varying numbers and identities of candidate routes. By running experiments on real datasets, the proposed approach is validated and compared against other commonly used assignment methods in the logistics industry. The results are quite promising: in the majority of the cases, PPO-OPA obtains similar performance to the primal dual method, but with a weaker assumption that the total parcel volume is not given. Finally, it is noteworthy that our approach actually provides a general framework that can be applied to any other similar online assignment/allocation problems by specifying an appropriate Online Assignment MDP.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In *International conference on machine learning*. PMLR, 22–31.

[2] Tobias Achterberg. 2009. SCIP: solving constraint integer programs. *Mathematical Programming Computation* 1, 1 (2009), 1–41.

[3] Shipra Agrawal, Erick Delage, Mark Peters, Zizhuo Wang, and Yinyu Ye. 2009. A unified framework for dynamic pari-mutuel information market design. In *Proceedings of the 10th ACM conference on Electronic commerce*. 255–264.

[4] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Seffi Naor. 2004. A general approach to online network optimization problems. In *Symposium on Discrete Algorithms: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, Vol. 11. 577–586.

[5] Santiago Balseiro, Christian Kroer, and Rachitesh Kumar. 2022. Online Resource Allocation under Horizon Uncertainty. *arXiv preprint arXiv:2206.13606* (2022).

[6] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. 2020. Dual mirror descent for online allocation problems. In *International Conference on Machine Learning*. PMLR, 613–628.

[7] Niv Buchbinder and Joseph Naor. 2006. Improved bounds for online routing and packing via a primal-dual approach. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 293–304.

[8] Niv Buchbinder, Joseph Seffi Naor, et al. 2009. The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science* 3, 2–3 (2009), 93–263.

[9] Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 311–320.

[10] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research* 18, 1 (2017), 6070–6120.

[11] Nikhil R Devanur and Thomas P Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*. 71–78.

[12] Bingqian Du, Chuan Wu, and Zhiyi Huang. 2019. Learning resource allocation and pricing for cloud profit maximization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7570–7577.

[13] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2009. Online Markov decision processes. *Mathematics of Operations Research* 34, 3 (2009), 726–736.

[14] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*.

[15] Chien-Ju Ho and Jennifer Vaughan. 2012. Online task assignment in crowdsourcing markets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26. 45–51.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[18] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2007. Adwords and generalized online matching. *Journal of the ACM (JACM)* 54, 5 (2007), 22–es.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

[21] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[23] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* (2017).

[24] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[25] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[26] Gerald Tesauro et al. 2005. Online resource allocation using decompositional reinforcement learning. In *AAAI*, Vol. 5. 886–891.

[27] Gerald Tesauro, Nicholas K Jong, Rajarshi Das, and Mohamed N Bennani. 2006. A hybrid reinforcement learning approach to autonomic resource allocation. In *2006 IEEE International Conference on Autonomic Computing*. IEEE, 65–73.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[29] Zhiyuan Xu, Yanzhi Wang, Jian Tang, Jing Wang, and Mustafa Cenk Gursoy. 2017. A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[30] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Fred Juang. 2019. Deep reinforcement learning based resource allocation for V2V communications. *IEEE Transactions on Vehicular Technology* 68, 4 (2019), 3163–3173.

[31] Qixin Zhang, Wenbing Ye, Zaiyi Chen, Haoyuan Hu, Enhong Chen, and Yang Yu. 2021. Online Allocation with Two-sided Resource Constraints. *arXiv preprint arXiv:2112.13964* (2021).

[32] Wei Zhang and Thomas G Dietterich. 1995. A reinforcement learning approach to job-shop scheduling. In *IJCAI*, Vol. 95. Citeseer, 1114–1120.