Deep Reinforcement Learning-based Scheduling for Same Day Delivery with a Dynamic Number of Drones

Boyang Zhou and Liang Cheng

[1] Lehigh University
[2] University of Toledo

Abstract. Same-Day Delivery (SDD) has emerged as a popular trend in the retail market, relieving workers from repetitive and monotonous tasks. Despite these advantages, SDD scheduling is challenging as there is no prior information available for upcoming tasks. Existing research has attempted to address this problem using local heuristic search, approximate dynamic programming, and reinforcement learning algorithms. However, none of these approaches has considered a dynamic number of drones, which can change due to unforeseen crashes or employing new drones due to the heavy workload. In this paper, we propose a Same-Day Delivery with a Dynamic Number of Drones (SD4) problem. To address this problem, we present a reinforcement learning model using Double Deep-Q Network (DDQN) to handle both task scheduling with a dynamic number of drones and drone employment simultaneously.

Keywords: Unmanned Aerial Vehicles (UAV) · Double Deep-Q Networks (DDQN).

1 Introduction

The advancement of battery technology and control theory has made Unmanned Aerial Vehicles (UAVs), or drones, more applicable to commercial settings. Same-day delivery (SDD) is one of the most important applications of UAVs. Although SDD is widely investigated by researchers in logistics [1–4, 6], existing research lacks the consideration of a changing number of vehicles (e.g., drones) during the delivery. Drones are vulnerable to crashes in harsh and uncertain environments and can be shared by different depots causing a changing number of drones. Thus, it is necessary to develop scheduling algorithms that deal with the dynamic number of drones. We propose a Same-Day Delivery with a Dynamic number of Drones (SD4) problem, specifically targeting same-day meal delivery.

[2] This work is supported by NSF Award No. 2146968. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the sponsors of the research.

The SD4 problem involves two main novelties: The first novelty is scheduling tasks for a dynamic number of drones, while the second novelty involves coordinating drones, including their deployment and release, to enable more efficient drone utilization, particularly when depots experience peak periods in different time periods. The main contributions of the paper are as follows:

1. We propose the SD4 problem, which considers the dynamic number of drones and drone coordination during delivery for more realistic delivery scenarios.
2. We use a reinforcement learning model that employs Double Deep-Q Networks (DDQN [5]) to solve the SD4 problem. This model is evaluated in terms of convergence, solution quality, and real-time performance.

The rest of the paper is organized as follows. Section 2 proposes the SD4 system for same-day meal delivery. Section 3 and Section 4 demonstrate the reinforcement learning model and environment setup. Section 5 evaluates our solution.

2 System Description for Meal Delivery with SD4

This section provides an overview of the SD4 system. We take same-day meal delivery as the application scenario of SD4. Figure 1 illustrates the architecture of the SD4 system for meal delivery.

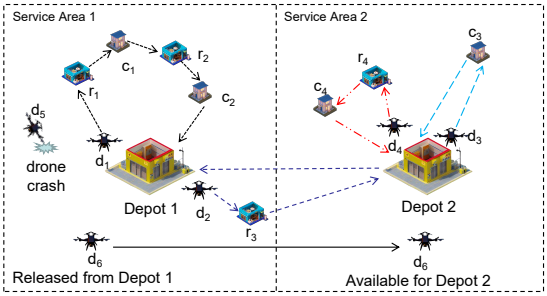

Fig. 1. The system architecture for the same-day meal delivery

A meal delivery task t_j in the SD4 system consists of two components: the restaurant and the customer, denoted as r_j and c_j, respectively, in Figure 1. There are two types of delivery in the system: individual delivery and collaborative delivery. Individual delivery occurs when a task can be assigned to a single drone. For instance, in Figure 1, tasks t_1 and t_2 are assigned to drone d_1 as individual deliveries. Collaborative delivery, on the other hand, is necessary when the restaurant and the customer of a task are located in different service areas of depots. Task t_3 is an example of collaborative delivery. Drones may crash during delivery. Moreover, depots may release their drones (i.e., d_6). Depot 2 must estimate whether employing d_6 would improve overall system performance after it is released by depot 1.

3 Environment setup for SDD task scheduling

Environment setup plays a crucial role in reinforcement learning. The environment of a reinforcement learning model consists of state, action, and reward. In this section, we introduce the environment setup for SDD task scheduling. We make the following assumptions for the SD4 problem:

1. Drones have a fixed maximum flight time regardless of the payload of a task.
2. Each drone can only deliver a single meal at a time.
3. Depots can reject any request without penalty, except in the case of a collaborative delivery where a drone has already picked up the meal.
4. Drones switch their batteries in depots and the switching time is negligible.

3.1 Task and State

Task t_j can be described by a five-tuple (arr_j, r_j, c_j, dl_j, p_j). arr_j is the arrival time of task t_j. r_j and c_j are 2D coordinates for the restaurant and the customer, respectively. dl_j is the deadline for finishing this task. p_j is the penalty for t_j. State S_j for task t_j in the task scheduling environment is a vector that contains the following elements:

1. $Avl = [avl_1, avl_2, ..., avl_n]$ indicating the ready time for the next departure of each drone, where n is the number of drones in the system.
2. p_j and dl_j, representing the penalty and deadline for task t_j, respectively, which are appended to S_j.
3. $F_j = [f_{1j}, f_{2j}, ..., f_{nj}]$, where f_{ij} is the time for the drone i to execute t_j.

3.2 Action and Reward

Action a_j needs to be taken for task t_j. a_j can take one of the following values:

$$a_j = \begin{cases} 0 & \text{if } t_j \text{ is rejected} \\ k & t_j \text{ is assigned to the } k\text{th drone, } 1 \leq k \leq n \end{cases} \tag{1}$$

The immediate reward $R(S_j, a_j)$ is designed to provide feedback action a_j under the current state S_j. It is defined by Equation 2.

$$R(S_j, a_j) = \begin{cases} p_j & \text{if } t_j \text{ is rejected} \\ 1 & tf_{a_j} <= dl_j \\ 1 - \frac{tf_{a_j} - dl_j}{t_{max}} & \text{Otherwise} \end{cases} \tag{2}$$

4 Reinforcement Learning Model and Environment Setup for Drone Employment

This section discusses the reinforcement learning model and the environment setup for the admission of drones. Figure 2 depicts the model for drone employment, where the shared environment can adaptively choose the agent and provide states and rewards based on the task type. The $DDQN_2$ set contains trained DDQN models for scheduling with different numbers of drones.

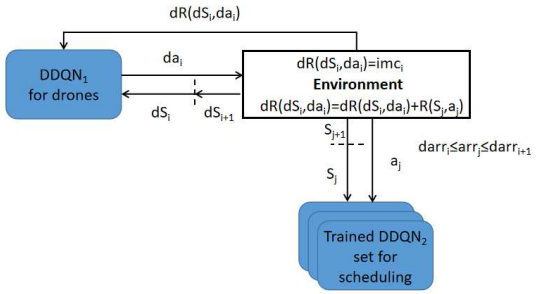

Fig. 2. Reinforcement learning model for drone employment

4.1 Task and State

A two-tuple ($darr_i$, $davl_i$) describes employable drone dt_i, where $darr_i$ is the time when the depot is notified of the availability of dt_i, and $davl_i$ is the time when dt_i will be ready for the next task. dS_i is the state for drone employment at $darr_i$. It includes the following elements:

1. $Avl' = [avl_1,...,avl_{maxN}]$, where $maxN$ is the maximum number of drones accommodated by the depot. When $i > n$, then $avl_i = t_{max}$ so that no work can be assigned to the ith drone since it is absent.
2. $davl_i$ notifies the agent of the time when dt_i will be ready for meal delivery.
3. $End = [end_1, ..., end_{maxN}]$ indicates the off-work time for each drone. Drone i will stop working and leave the depot at end_i. N represents the number of non-employed drones belonging to the depot. Drones belonging to the depot are not allowed to leave the depot unless they crash.

4.2 Action and Reward

Two decisions must be made for each employable drone: whether to accept it and the corresponding release time if accepted. Hence, da_i is designed to make these two simultaneous decisions. The values of da_i are as follows:

$$da_i = \begin{cases} 0 & dt_i \text{ is rejected} \\ k & dt_i \text{ is accepted and it needs to work for } \frac{t_{max}*k}{l},\ 1 \le k \le l \end{cases} \tag{3}$$

The parameter l is used to divide the time horizon t_{max} into l equal-length time intervals. $dR(dS_i, da_i)$ is the reward when action da_i is taken under state dS_i. $dR(dS_i, da_i)$ contains a immediate cost imc_i of employing drone dt_i. imc_i is calculated using dp, $davl_i$, and da_i in Equation 4.

$$imc_i = \begin{cases} 0 & \text{if } da_i = 0 \\ (\min(t_{max}, davl_i + \frac{t_{max}*da_i}{l}) & \\ -davl_i) * \frac{dp}{t_{max}} & \text{Otherwise} \end{cases} \tag{4}$$

Besides imc_i, $dR(dS_i, da_i)$ should also incorpotate the reward for task scheduling. Thus, $dR(dS_i, da_i)$ can be calculated using Equation 5.

$$dR(dS_i, da_i) = imc_i + \sum_j R(S_j, a_j) \tag{5}$$

, where $darr_i < arr_j < darr_{i+1}$.

5 Evaluation

This section presents the evaluation of our reinforcement learning-based approach for SD4, which is divided into two parts: (i) evaluation for SDD task scheduling, and (ii) evaluation for the model for drone employment.

5.1 Evaluation of DDQN for SDD Task Scheduling

We use simulations to evaluate the performance of our proposed model. Here is the setup. The service area of a depot is a 30 by 30 plane, where each unit distance takes one minute for a drone to travel. Euclidean distance is used for the calculation. The maximum shift duration (t_{max}) is set to 600 minutes, and the maximum flight time $maxF$ of a drone is 60 minutes. The r_j and c_j coordinates are randomly generated, ensuring that the drone can complete tasks in an individual way or a collaborative way within $maxF$. The penalty p_j is set to 0 for individual delivery tasks and -1 for collaborative ones with food picked up.

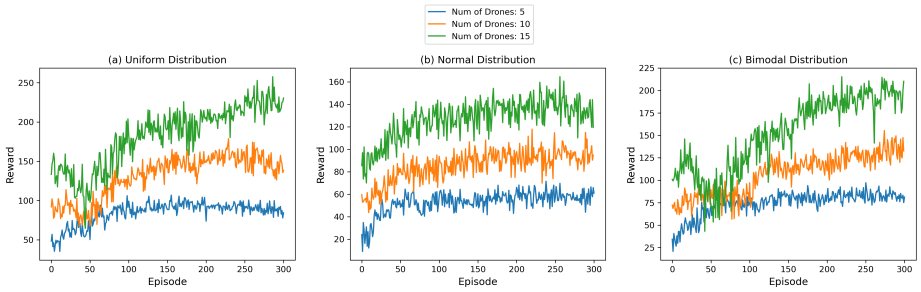

Fig. 3. Plots for the rewards vs. training episodes using 5, 10, 15 drones under different distributions

Convergence Evaluation To investigate how the workload distribution affects the performance of DDQN, we generated workload using uniform, normal, and bimodal distributions. Figure 3 shows the variation of SDD meal delivery rewards with trained episodes for 5, 10, and 15 drones under the three workload distributions. Each episode represents a shift in the system. As we can see, the model converges well in all scenarios.

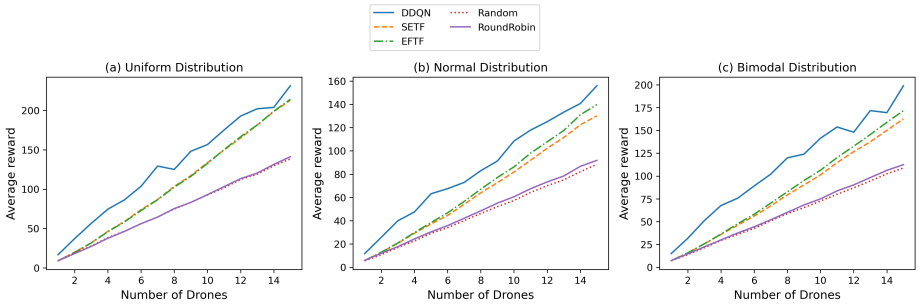

Fig. 4. The average rewards and the average number of scheduled tasks achieved by DDQN, greedy algorithm 1, and greedy algorithm 2 with different numbers of drones under the three workload distributions

Solution Quality Evaluation To evaluate the quality of solutions of our DDQN, we compare our DDQN model with some traditional scheduling algorithms, including the Shortest Execution Time First (SETF), the Earliest Finish Time First (EFTF), round-robin, and random selection.

SETF: The depot greedily assigns task t_j to the drone that has the shortest task completion time (i.e., $a_j = \underset{i}{argmin}\, f_{ij}$).

EFTF: In EFTF, instead of greedily choosing the shortest execution time, the depot assigns the task t_j to the drone, which has the smallest avl_i after tasking the task (i.e., $a_j = \underset{i}{argmin}\, (avl_i + f_{ij})$.

The evaluation for solution quality was conducted by running 100 episodes for each scenario, and then calculating the average rewards of five algorithms. Figure 4 (a), (b), and (c) shows the average rewards versus different numbers of drones achieved by the five algorithms under different workload distributions.

Three workload distributions were used in the evaluation, and all three distributions have the same workload expectation. DDQN can increase the average reward by a range from 2.5% to 104.1% compared with the best result from the four traditional real-time scheduling algorithms.

5.2 Evaluation for the Drone Employment Model

In this section, we mainly evaluate the model for drone employment. The service area and task generation are inherited from Section 5.1. A depot is assumed to have 4 drones initially, and $maxN$ is 8. We evaluate the performance of DDQN for employable drones under normal and bimodal distributions. The uniform distribution is not included as it cannot benefit from employing drones. There will be 20 employable drone requests uniformly distributed in each shift. l is set to 10 for the selection of da_i and the calculation of end_i.

Convergence Evaluation The first part of the evaluation focuses on the convergence of the DDQN model for employable drones. We set dp to -20 in this

example. The rewards vs. training episodes for normal and bimodal workload distributions are shown in Figure 5, with the mean rewards calculated for the closest 50 episodes to provide a better measurement of convergence. The results indicate that the DDQN model for employable drones converges well in both distributions. It takes fewer than 200 episodes for the model to converge under the two workload distributions.

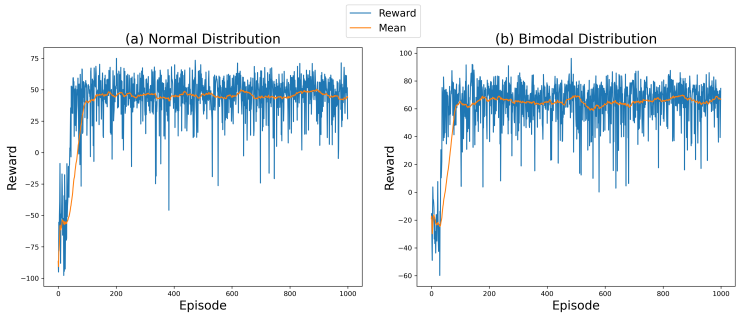

Fig. 5. Plots for the rewards vs. training episodes of DDQN for employable drones under different distributions

Solution Quality Evaluation The solution quality evaluation involves comparing the performance of the drone employment model to that of DDQN models designed solely for SDD task scheduling under different dp values. We use five DDQN models that can schedule tasks using 4 to 8 drones as baselines. To ensure a fair comparison, we adjust the reward for the DDQN models for task scheduling by adding $dp * (n - 4)$, where n is the number of drones. We then compare the gap between the reward obtained from the drone employment model and the best-adjusted reward derived from the five DDQN models under different dp values to evaluate the solution quality.

Figure 6 presents the reward achieved by the drone employment model and the reward gap in different scenarios. As $|dp|$ increases, the reward gap and reward tend to decrease. There is an outlier of the reward gap when $|dp|$ is 5 under the normal workload distribution. In this case, an additional drone can always produce a positive effect because $|dp|$ is small. However, the decrement stops when $|dp|$ reaches a threshold in both distributions since employing a drone with a high cost will always cause a negative effect. Thus, the model stops employing any drone in these scenarios, causing a 0 reward gap and a constant reward. Drone employment in SD4 will not cause a negative effect no matter how large $|dp|$ is and can increase the reward by up to 21.0% compared with the best-adjusted reward from DDQN models for static numbers of drones.

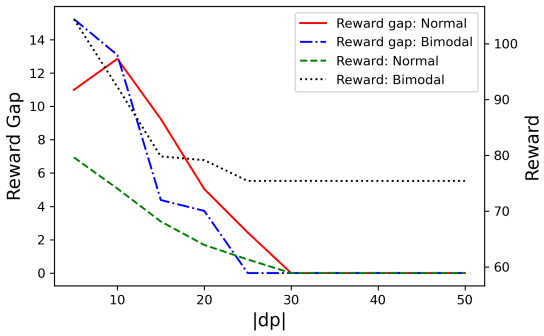

Fig. 6. Reward gap between the drone employment model and DDQN models

5.3 Real-time Performance Evaluation

The evaluation is conducted on a machine equipped with an i9-13900k CPU and an Nvidia RTX 4080 GPU. The average inference time is less than 50 μs for both models in all scenarios.

6 Conclusion

This paper introduces a novel approach to solving the SD4 problem using a reinforcement learning model. The proposed model is capable of dynamic task scheduling and employment of drones by depots, which can increase the overall system efficiency. The model is evaluated in terms of convergence, solution quality, and real-time performance. The results show that the DDQN model for task scheduling outperforms traditional greedy algorithms under different workload distributions and numbers of drones, while the drone employment model further enhances the system efficiency.

References

1. Dayarian, I., Savelsbergh, M.: Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. Production and Operations Management **29**(9), 2153–2174 (2020)
2. Dayarian, I., Savelsbergh, M., Clarke, J.P.: Same-day delivery with drone resupply. Transportation Science **54**(1), 229–249 (2020)
3. Klapp, M.A., Erera, A.L., Toriello, A.: The dynamic dispatch waves problem for same-day delivery. European Journal of Operational Research pp. 519–534 (2018)
4. Schubert, D., Kuhn, H., Holzapfel, A.: Same-day deliveries in omnichannel retail: Integrated order picking and vehicle routing with vehicle-site dependencies. Naval Research Logistics (NRL) **68**(6), 721–744 (2021)
5. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI conference on artificial intelligence (2016)
6. Voccia, S.A., Campbell, A.M., Thomas, B.W.: The same-day delivery problem for online purchases. Transportation Science **53**(1), 167–184 (2019)