UAV parcel Delivery System with Deep Reinforcement Learning Based Collision Avoidance and Route Optimization

基于深度强化学习的无人机包裹配送系统，具有碰撞避免和路线优化功能

Chun-Yuan Chi1, De-Fu Chen2, Hoang-Phuong Doan2, and Chung-Hsien Kuo2, *

Chun-Yuan Chi1, De-Fu Chen2, Hoang-Phuong Doan2, 和 Chung-Hsien Kuo2, *

[1] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

[1] 台湾科技大学电机工程系, 台北, 台湾

[2] Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan

" [2] 台湾大学机械工程系，台北，台湾"

*Corresponding author: chunghsien@ntu.edu.tw

"* 通讯作者:chunghsien@ntu.edu.tw"

(Received 28 June 2024, Accepted 09 July 2024, Published Online 16 July 2024)

"(收到日期 2024 年 6 月 28 日，接受日期 2024 年 7 月 9 日，在线发表日期 2024 年 7 月 16 日)"

This work was supported by the National Science and Technology Council, Taiwan under the Grant NSTC 112-2221-E-

本研究得到了台湾国家科学技术委员会的支持，资助编号 NSTC 112-2221-E-

011-140. (Corresponding author: Chung-Hsien Kuo).

011-140. (通讯作者: 郭忠贤)。

DOI: https://doi.org/10.5875/p25fht10

DOI: https://doi.org/10.5875/p25fht10

Abstract: Unmanned Aerial Vehicles (UAVs), or drones, have recently become a favorable solution for fast parcel delivery due to their maneuverability and advances in navigation technologies. With the limitation of battery capacity and payload of drones, it is crucial to consider both efficiency and cost while conducting the tasks. Meanwhile, UAVs should not collide with each other while traveling to customers. In this article, we propose a UAV parcel delivery system involving deep reinforcement learning (DRL) approach for collision avoidance and a genetic algorithm for route optimization. Specifically, a delivery center generates near-optimal routes, loading UAV with parcels according to demands. Each UAV takes charge of delivering packages in compliance with the assigned route while avoiding collision with each other. We utilize DRL to achieve collision avoidance without having prior knowledge about the trajectories of other UAVS. Additionally, we adopt a genetic algorithm to obtain the lowest energy cost path for each UAV. To find such an optimized path, we solve a capacitated vehicle routing problem (CVRP) with a modified cost function and extra constraints. Realistic simulations using a physics engine and software-in-the-loop (SITL) are conducted to evaluate the

摘要: 无人驾驶飞行器 (UAVs)，或称为无人机，由于其机动性和导航技术的进步，最近成为快速包裹递送的一种受欢迎的解决方案。由于无人机的电池容量和载重限制，执行任务时必须同时考虑效率和成本。同时，无人机在飞往客户的途中不应相互碰撞。在本文中，我们提出了一个涉及深度强化学习 (DRL) 方法的无人机包裹递送系统，用于避碰，以及使用遗传算法进行路线优化。具体来说，配送中心根据需求生成近优路线，并加载包裹到无人机上。每架无人机负责按照指定的路线递送包裹，同时避免与其他无人机相撞。我们利用 DRL 实现避碰，无需了解其他无人机的轨迹。此外，我们采用遗传算法为每架无人机获取最低能耗路径。为了找到这样的优化路径，我们解决了一个带有修改后的成本函数和额外约束的容量车辆路径问题 (CVRP)。使用物理引擎和软件在环 (SITL) 进行了真实模拟，以评估

Keywords: Capacitated vehicle routing problem, Genetic Algorithm, Reinforcement Learning, UAV collision avoidance.

关键词: 容量车辆路径问题，遗传算法，强化学习，无人机避碰。

I. Introduction

I. 引言

Unmanned Aerial Vehicles (UAVs) are recently being explored due to their potential to be adopted in parcel delivery systems. With the growth of demands in the logistics and e-commerce industry, many logistics companies have shown great interest in using UAVs for delivery since parcels can be delivered more efficiently. Amazon Prime Air has become a pioneer in this field, expecting to adopt drones to replace delivery trucks, making the delivery faster compared to the present day [1]. This parcel delivery scheme is called the drone-only scheme, where the drone departs from the depot, delivers goods to the customer, and then returns to the depot. This scheme suffers from the distance limitation due to the battery capacity of the drone. Several publications proposed methods to expand the coverage area by adding charge stations and warehouses in the network [2], [3], [4].

无人驾驶飞行器 (UAVs) 最近因其被应用于包裹递送系统的潜力而受到关注。随着物流和电子商务行业需求的增长，许多物流公司表现出对使用 UAVs 进行递送的极大兴趣，因为包裹可以更高效地递送。Amazon Prime Air 已成为这一领域的先驱，期望采用无人机替代递送卡车，使得递送速度比目前更快 [1]。这种包裹递送方案被称为仅无人机方案，其中无人机从仓库出发，将货物递送给客户，然后返回仓库。这种方案受到无人机电池容量限制的距离限制。一些出版物提出了通过在网络中添加充电站和仓库来扩大覆盖区域的方法 [2]、[3]、[4]。

In recent years, another scheme, called the Truck-and-Drone scheme or last-mile drone delivery, has drawn much attention in the research community. The Truck-and-Drone scheme expands the delivery coverage and improves energy efficiency by combining heterogeneous vehicles. To coordinate drones with trucks, Wang et al. proposed a heuristic routing and scheduling algorithm to solve the hybrid parcel delivery problem [5]. Nirupam Das et al. synchronized drones and delivery trucks by developing a multi-objective optimization model that minimizes travel costs and maximizes customer service level in terms of timely deliveries [6]. Chen et al. developed a drone delivery system that achieves mixed indoor- outdoor autopilot operation [7].

近年来，另一种被称为卡车-无人机方案或最后一英里无人机递送的方案在研究界引起了广泛关注。卡车-无人机方案通过结合异构车辆扩大了递送覆盖范围并提高了能源效率。为了协调无人机与卡车，Wang 等人提出了一种启发式路由和调度算法来解决混合包裹递送问题 [5]。Nirupam Das 等人通过开发一个多目标优化模型同步无人机和递送卡车，该模型最小化了旅行成本并在及时递送方面最大化了客户服务水平 [6]。Chen 等人开发了一个无人机递送系统，实现了室内-室外混合自动飞行操作 [7]。

For a system with multiple UAVs, it is crucial to consider the risk of collision among them. It is difficult for each UAV to avoid others without a centralized control mechanism. Therefore, we propose a decentralized collision avoidance method by adopting a DRL-based approach. Additionally, each UAV has a payload limit which limits the weight and quantity of parcels the drone can carry. The sequence of how UAVs travel to customers becomes critical since the gross weight of the UAV will directly impact battery drain and, furthermore, affect the logistics cost. As a solution, we aim to minimize the total energy consumed by UAVs while traveling instead of minimizing total travel distance like the traditional capacitated vehicle routing problem (CVRP). For optimization, we adopt a genetic algorithm with a custom fitness function.

对于一个包含多个无人机的系统，考虑它们之间的碰撞风险是至关重要的。在没有集中控制机制的情况下，每个无人机避免与其他无人机相撞是困难的。因此，我们提出了一种基于深度强化学习 (DRL) 的分布式碰撞避免方法。此外，每个无人机都有载重限制，这限制了无人机可以携带的包裹的重量和数量。无人机前往客户的顺序变得关键，因为无人机的总重量将直接影响电池消耗，进而影响物流成本。作为解决方案，我们的目标是在无人机行驶过程中最小化总能耗，而不是像传统的容量车辆路径问题 (CVRP) 那样最小化总行驶距离。为了优化，我们采用了一种遗传算法，并配备了一个自定义的适应度函数。

In this article, we demonstrate the feasibility of the proposed collision avoidance method by conducting realistic simulations in a physics engine with SITL flight controller simulation on Robot Operating System (ROS) [8]. We also simulate the entire UAV parcel delivery process in a campus environ- ment, including loading packages, traveling to customers, and dropping packages. The contributions of this article can be summarized as follows:

在本文中，我们通过在物理引擎中进行现实模拟，并在机器人操作系统 (ROS) 上使用 SITL 飞行控制器模拟，证明了所提出碰撞避免方法的可行性。我们还模拟了在校园环境中完整的无人机包裹配送过程，包括装载包裹、前往客户处以及投放包裹。本文的贡献可以概括如下:

- An application of DRL-based obstacle avoidance method to a parcel delivery system consisting of multiple UAVs with Soft Actor Critic Framework.

- 将基于深度强化学习的障碍物避免方法应用于由多个无人机组成的包裹配送系统，采用软行为者评判家框架。

- The usage of a genetic algorithm to generate suitable routes for multiple UAVs in the system based on total energy consumption.

- 使用遗传算法为系统中的多个无人机生成基于总能耗的合适路线。

The rest of this article is organized as follows: Section II presents related work. Section III states the system architecture and the problem definitions. Section IV addresses the UAV collision avoidance problem with a state-of-the-art deep reinforcement learning algorithm. Section V discusses the route optimization problem for capacitated vehicles in UAV parcel delivery tasks. Section VI carries out the validation and simulation results. Section VII concludes the article.

本文的其余部分安排如下: 第二部分介绍相关工作。第三部分阐述系统架构和问题定义。第四部分使用最先进的深度强化学习算法解决无人机碰撞避免问题。第五部分讨论无人机包裹配送任务中容量车辆的路线优化问题。第六部分进行验证和模拟结果分析。第七部分总结全文。

II. Related Works

II. 相关工作

A.UAV Collision Avoidance

A. 无人机碰撞避免

Collision avoidance is a fundamental requirement for multi-UAV systems. Centralized collision models for UAVs are presented. Loayza et al. proposed a centralized model providing feedback in real-time to the agents while considering trajectory calculation and collision avoidance [9]. Mellinger et al. presented an algorithm for the generation of optimal trajectories for teams of heterogeneous quadrotors in three-dimensional environments with obstacles by formulating the problem us- ing mixed-integer quadratic programs (MIQPs) [10]. Both solutions rely on a central server communicating with every agent and generating global control commands according to the observations for all UAVs. However, implementing such centralized systems in large environments is usually difficult since it heavily relies on the communication between agents and the server; delay or interference in signal transmissions may lead to unwanted results.

避碰是多无人机系统的基础要求。本文提出了针对无人机的集中式碰撞模型。Loayza 等人提出了一种集中式模型，该模型在考虑轨迹计算和避碰的同时，能够实时向代理提供反馈 [9]。Mellinger 等人提出了一种算法，用于在具有障碍物的三维环境中为异质四旋翼机队生成最优轨迹，通过使用混合整数二次规划 (MIQPs) 来构建问题 [10]。这两种解决方案都依赖于一个中心服务器与每个代理进行通信，并根据对所有无人机的观察生成全局控制命令。然而，在大型环境中实施这种集中式系统通常是困难的，因为它严重依赖于代理与服务器之间的通信；信号传输的延迟或干扰可能导致不希望的结果。

In recent years, researchers have turned their interest into decentralized methods. In such systems, agents take action based only on their own observations. Several works [11]- [13] have successfully adopted reinforcement learning to train policies to plan collision-free trajectories by leveraging local observations. However, the method proposed in [11] assumes all UAVs fly at the same speed and can only output dis- cretized turning angles, which may cause jerky movement. The off-policy actor-critic-based reinforcement learning algorithm, deep deterministic policy gradient (DDPG) [14], used in [12], is hyperparameter-sensitive [15] and suffers from finding the optimal policy due to its non-stochastic characteristic. Researchers in [13] formulated the UAV collision avoidance problem with wireless connectivity constraints as a Markov decision process (MDP) and optimized the value function of the MDP to find the optimal policy. However, the proposed method was not tested in a realistic environment, so one cannot guarantee the feasibility of the policy in an environment with noises or delays.

近年来，研究人员开始关注分布式方法。在这样的系统中，代理仅根据自身的观察采取行动。一些工作 [11]- [13] 已成功采用强化学习来训练策略，通过利用局部观察来规划无碰撞轨迹。然而，[11] 中提出的方法假设所有无人机以相同的速度飞行，并且只能输出离散的转向角，这可能导致动作生硬。在 [12] 中使用的基于演员-评论家策略的强化学习算法，深度确定性策略梯度 (DDPG)[14]，对超参数敏感 [15]，并且由于其非随机特性，难以找到最优策略。[13] 中的研究人员将具有无线连接约束的无人机避碰问题构建为马尔可夫决策过程 (MDP)，并优化了 MDP 的值函数以找到最优策略。然而，提出的方法并未在现实环境中进行测试，因此无法保证在带有噪声或延迟的环境中策略的可行性。

A method based on Multi-Agent Reinforcement Learn- ing is presented in [16] to conduct large-scale searching in an unknown environment with multi-UAVs. Furthermore, a new multi-agent recurrent deterministic policy gradient (MARDPG) algorithm is proposed in [17] to achieve the goal of obstacle avoidance for multi-UAVs. In [18], a Deep Reinforcement Learning-based collision avoidance algorithm with Attention-Based Policy Distillation is proposed, enabling UAVs to conduct obstacle avoidance more efficiently and accurately. Hui et al. discuss the use of a Decentralized Exploration Planning approach based on a lightweight in- formation structure for multi-UAV systems [19]. Although various approaches for tackling collisions among UAVs have been published, none were applied to parcel delivery systems, indicating significant potential for research in this field.

在文献 [16] 中，提出了一种基于多智能体强化学习的方法，用于在未知环境中使用多无人机进行大规模搜索。此外，在文献 [17] 中，为了实现多无人机的避障目标，提出了一种新的多智能体递归确定性策略梯度 (MARDPG) 算法。在文献 [18] 中，提出了一种基于深度强化学习的碰撞避免算法，并结合了基于注意力的策略蒸馏，使得无人机能够更高效、更精确地进行避障。Hui 等人讨论了基于轻量级信息结构的分布式探索规划方法在多无人机系统中的应用 [19]。尽管已经发表了各种解决无人机之间碰撞的方法，但它们均未应用于快递配送系统，这表明在这一领域的研究具有巨大的潜力。

B. Scheduling and Routing Problems for Drone Delivery

B. 无人机配送的调度和路由问题

As drone technology matures, many large organizations have shown interest in drone delivery. Even though significant efforts have been put into developing drone delivery technologies, the drone delivery planning problem poses a new challenge due to limited flight range and payload of drones. Traditional traveling salesman problems (TSP) or vehicle routing problems (VRP) are no longer adequate for formulating the drone delivery routing problem. Some variants, such as [20] and [21], proposed flying sidekick traveling salesman problem (FSTSP) and vehicle routing problem with drones (VRPD) respectively. The VRPD is considered an extension of the FSTSP. While the FSTSP considers only one drone and one truck in the entire operation, VRPD utilizes multiple trucks and drones to make deliveries while considering the capacity of both trucks and drones. However, both works ignored factors crucial to practical drone delivery, such as changing payload weights and energy costs. Yao et al. discovered a scheduling approach using an Evolutionary Utility Prediction Matrix, which can adapt to the environment dynamically [22]. However, precise tuning of parameters is needed.

随着无人机技术的成熟，许多大型组织对无人机配送表现出了兴趣。尽管在开发无人机配送技术方面已经投入了大量努力，但由于无人机的飞行范围和载重限制，无人机配送规划问题提出了新的挑战。传统的旅行商问题 (TSP) 或车辆路径问题 (VRP) 不再足以描述无人机配送路由问题。一些变种，如文献 [20] 和 [21] 分别提出的飞行助手旅行商问题 (FSTSP) 和带无人机的车辆路径问题 (VRPD)。VRPD 被视为 FSTSP 的扩展。FSTSP 仅考虑整个操作中的一个无人机和一辆卡车，而 VRPD 则利用多辆卡车和无人机进行配送，同时考虑卡车和无人机的载重能力。然而，这两项工作都忽略了实际无人机配送中的关键因素，如变化的载重和能源成本。Yao 等人发现了一种使用进化效用预测矩阵的调度方法，该方法能够动态适应环境 [22]。但是，需要精确调整参数。

Recently, several researchers have focused on reducing cost or energy consumption. Dorling et al. proposed solving drone delivery problems (DDPs) with a multi-trip VRP (MTVRP) [23]. They focused on minimizing cost or delivery time while considering battery weight, payload weight, and drone reuse. In [24], the authors focused on minimizing the total energy consumption in electric vehicle routing problems with drones (EVRPD). In this thesis, we consider a similar scenario with drones being the only vehicle in the system. We adopt the energy cost function in [24] and use it in a genetic algorithm.

最近，一些研究者专注于降低成本或能耗。Dorling 等人提出使用多行程车辆路径问题 (MTVRP) 解决无人机配送问题 (DDPs)[23]。他们关注在考虑电池重量、载重重量和无人机重复使用的情况下，最小化成本或配送时间。在 [24] 中，作者关注于最小化电动车辆路径问题中无人机的总能耗 (EVRPD)。在本论文中，我们考虑一个类似的场景，即无人机是系统中的唯一交通工具。我们采用 [24] 中的能耗成本函数，并将其用于遗传算法中。

III. Problem Definitions

III. 问题定义

We address two major problems in this article: UAV collision avoidance and optimized route solutions in the parcel delivery system. We first state the definitions of the two problems respectively, then delve into detailed content in the

本文解决了两个主要问题: 无人机避障和包裹配送系统中的优化路径解决方案。我们首先分别陈述这两个问题的定义，然后在接下来的章节中深入详细内容。

following sections.

以下各节。

A.UAV Collision Avoidance

A. 无人机避障

Consider an environment consisting of a set $\mathcal{M}=\{1,2,\ldots,m\}$ of m UAVs flying at a constant altitude. For each UAV to avoid collisions with others we must ensure the distance $d_{i,j}$ between each UAV is less than the minimum collision distance $d_{\min}$ as shown in (1).

考虑一个由一组 $\mathcal{M}=\{1,2,\ldots,m\}$ 的 m 无人机在恒定高度飞行组成的环境。为了确保每架无人机避免与其他无人机相撞，我们必须保证每架无人机之间的距离 $d_{i,j}$ 小于最小碰撞距离 $d_{\min}$ ，如 (1) 所示。

$$d_{i,j}<d_{\min}\forall i,j\in\mathcal{M},i\neq j \tag{1}$$

In this article, we also assume a UAV k can fly with arbitrarily speed v_k and heading angle θ_k where v_k and θ_k are continuous values instead of discrete values. Our goal here is to find an approach to make UAVs able to avoid each other or obstacle automatically, and the approach we adopt is Deep Reinforcement Learning (DRL)

在本文中，我们还假设无人机 k 可以以任意速度 v_k 和航向角 θ_k 飞行，其中 v_k 和 θ_k 是连续值而不是离散值。我们的目标是找到一种方法，使无人机能够自动避开彼此或障碍物，我们采用的方法是深度强化学习 (DRL)。

B. Routes Optimization

B. 路径优化

We consider a capacitated vehicle routing problem (CVRP) with both weight and volume constraints since UAVs have a limited payload and space to carry parcels. Instead of minimizing the traveling cost directly calculated with the traveled distance of vehicles in conventional CVRP, we aim to reduce the total energy cost since modern UAVs use batteries as power sources and the electricity consumption directly affects logistics costs. We define our energy cost function of UAV as follows, similar to [24].

我们考虑了一个带有限制条件的容量车辆路径问题 (CVRP)，包括重量和体积限制，因为无人机的有效载荷和装载包裹的空间有限。与传统的 CVRP 直接最小化基于车辆行驶距离的旅行成本不同，我们的目标是减少总能耗成本，因为现代无人机使用电池作为电源，电力消耗直接影响物流成本。我们定义了无人机的能耗成本函数，如下所示，与文献 [24] 类似。With the definition of the cost function (2), our objective is to find a solution that minimizes this cost function. In this article, we apply a genetic algorithm to solve the optimization problem.

$$\text{Cost }_{\text{Energy }}=\text{ Distance }\times(1+\text{ Payload })\tag{2}$$

With the definition of cost function (2), our goal is to find a solution that minimize the cost function. In this article, we apply genetic algorithm to solve the optimization problem.

在定义了成本函数 (2) 之后，我们的目标是找到一个最小化该成本函数的解决方案。在本文中，我们应用遗传算法来解决这个问题。### IV.DRL BASED UAV COLLISION AVOIDANCE

IV.DRL BASED UAV COLLISION AVOIDANCE

IV. 基于深度强化学习的无人机避障 We propose an approach for UAV collision avoidance using a state-of-the-art deep reinforcement learning algorithm called Soft Actor-Critic (SAC) [25]. Specifically, SAC is an off-policy actor-critic deep reinforcement learning algorithm based on the maximum entropy reinforcement learning framework. Unlike Deep Deterministic Policy Gradient (DDPG) [26] or Twin Delayed Deep Deterministic Policy Gradient (TD3) [27], SAC employs a stochastic policy with entropy regularization instead of a deterministic policy.

We propose an approach for UAV collision avoidance using a state-of-the-art deep reinforcement learning algorithm called Soft Actor-Critic (SAC) [25]. Specifically, SAC is an off-policy actor-critic deep reinforcement learning algorithm based on the maximum entropy reinforcement learning framework. Unlike Deep Deterministic Policy Gradient (DDPG) [26] or Twin Delayed Deep Deterministic Policy Gradient (TD3) [27], SAC uses a stochastic policy with entropy regularization instead of a deterministic policy.

我们提出了一种基于最先进的深度强化学习算法——Soft Actor-Critic(SAC)[25] 的无人机避障方法。具体来说，SAC 是一种基于最大熵强化学习框架的离策略 actor-critic 深度强化学习算法。与深度确定性策略梯度 (DDPG)[26] 或双延迟深度确定性策略梯度 (TD3)[27] 不同，SAC 使用带有熵正则化的随机策略，而不是确定性策略。## A. State Space

A. State Space

A. 状态空间

Since method proposed only consider local observation of the agent, the state representation would be represented in the local coordinate system of UAV. Consider a tagged UAV agent in Fig. 1, the x -axis is the current heading of the agent, d_{goal} is the relative distance of the agent’s goal, d_{obs} is the relative distance of between the detected obstacle UAV and the tagged agent, θ_{goal} is the angle between the agent’s goal and the x -axis, θ_{obs} is the angle between the detected obstacle and the x -axis, ψ_{obs} is the heading of the obstacle UAV with respect to the x-axis. It is assumed that all UAVs fly toward their heading direction and all the angles and headings are within the range between $-\pi$ and $+\pi$.

由于提出的方法仅考虑了代理的局部观测，状态表示将在无人机的局部坐标系中表示。考虑图 1 中的标记无人机代理，x 轴是代理当前的航向，d_{goal} 是代理目标的相对距离，d_{obs} 是检测到的障碍无人机与标记代理之间的相对距离，θ_{goal} 是代理目标与 x 轴之间的角度，θ_{obs} 是检测到的障碍与 x 轴之间的角度，ψ_{obs} 是障碍无人机相对于 x 轴的航向。假设所有无人机都朝其航向飞行，所有角度和航向都在 $-\pi$ 和 $+\pi$ 之间的范围内。

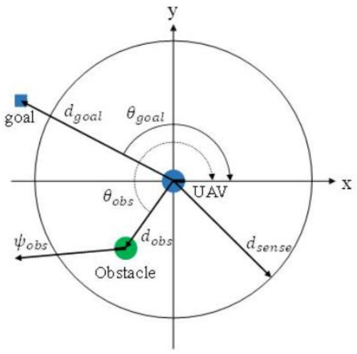

Fig. 1. The local state representation of a UAV agent.

图 1. 无人机代理的局部状态表示。

Let $\mathbf{s}_t$ be the current observed state of the tagged UAV at time t where $t \in [0, \infty)$. $\mathbf{s}_t$ is a composition of agent information info o ${}_{\text{agent}}$ and two nearest obstacle information info ${}_{obs1}$ and info ${}_{obs2}$.

设 $\mathbf{s}_t$ 为在时间 t 观测到的标记无人机当前状态，其中 $t \in [0, \infty)$ 。$\mathbf{s}_t$ 是由代理信息 info o ${}_{\text{agent}}$ 和两个最近障碍信息 info ${}_{obs1}$ 及 info ${}_{obs2}$ 组成。

$$\mathbf{s}_t = \left[\begin{array}{lll} \text{info}_{\text{agent}} & \text{info}_{\text{obs1}} & \text{info}_{\text{obs2}} \end{array} \right] \tag{3}$$

where info ${}_{\text{agent}}$ includes the information of the current agent speed v_{agent} , current heading angle ψ_{agent} with respect to the world coordinate system, relative goal distance dgoal, and θ_{goal} the angle between the agent' s goal and the x -axis. To keep state values in the same order of magnitude, each term in info ${}_{\text{agent}}$ is divided by a constant denominator where $v_{\max}$ is the maximum velocity of all UAVs in the system and d_{scale} is the predefined distance normalization factor.

其中 info ${}_{\text{agent}}$ 包括当前代理速度 v_{agent} 的信息，相对于世界坐标系的当前航向角 ψ_{agent} ，相对目标距离 dgoal，以及 θ_{goal} 代理目标与 x 轴之间的角度。为了保持状态值在同一数量级上， info ${}_{\text{agent}}$ 中的每一项都除以一个常数分母，其中 $v_{\max}$ 是系统中所有无人机的最大速度，d_{scale} 是预定义的距离归一化因子。

$$\text{info}_{\text{agent}} = \left[\begin{array}{llll} \frac{v_{\text{agent}}}{v_{\max}} & \frac{\psi_{\text{agent}}}{\pi} & \min\left(1, \frac{d_{\text{goal}}}{d_{\text{scale}}}\right) & \frac{\theta_{\text{goal}}}{\pi} \end{array} \right] \tag{4}$$

info ${}_{obs}$ represents the information of the obstacle UAV, including relative distance d_{obs} , relative angle θ_{obs} and ψ_{obs} the heading of the obstacle UAV. Likewise, each term in info ${}_{obs}$ is divided by a constant denominator.

info ${}_{obs}$ 表示障碍无人机信息，包括相对距离 d_{obs} 、相对角度 θ_{obs} 以及障碍无人机的航向 ψ_{obs} 。同样， info ${}_{obs}$ 中的每一项都除以一个常数分母。

$$\text{info}_{\text{obs}} = \left[\begin{array}{lll} \frac{d_{obs}}{d_{sense}} & \frac{\theta_{obs}}{\pi} & \frac{\psi_{obs}}{\pi} \end{array} \right] \tag{5}$$

It is worth noticing that $\text{info}_{obs} = \left[\begin{array}{lll} 1 & 1 & 0 \end{array} \right]$ if the tagged UAV cannot detect any UAV at the time.

值得注意的是 $\text{info}_{obs} = \left[\begin{array}{lll} 1 & 1 & 0 \end{array} \right]$ ，如果标记的无人机在此时无法检测到任何无人机。

B. Action Space

B. 动作空间

We model the quadrotor UAVs as a unicycle model as in Fig. 2. The velocity of a UAV moving in the x -axis is given by $v_x = v \cos \theta$ and the velocity of a UAV moving in the y -axis is given by $v_y = v \sin \theta$. Let $\mathbf{a}_t$ denotes the action space, $v \in [0, v_{\max}]$ be the UAV flying velocity and $\theta \in [-\pi, \pi]$ be the heading angle of the UAV with respect to the world coordinate system.

我们将四旋翼无人机 (UAVs) 建模为如图 2 所示的独轮车模型。无人机在 x 轴上的速度由 $v_x = v \cos \theta$ 给出，而在 y 轴上的速度由 $v_y = v \sin \theta$ 给出。令 $\mathbf{a}_t$ 表示动作空间，$v \in [0, v_{\max}]$ 为无人机的飞行速度，$\theta \in [-\pi, \pi]$ 为无人机相对于世界坐标系的航向角。

$$\mathbf{a}_t = \left[\frac{2v}{v_{\max}} - 1 \; \frac{\theta}{\pi} \right] \tag{6}$$

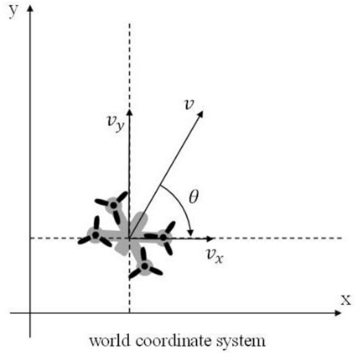

Fig. 2. Unicycle model of a quadrotor UAV.

图 2. 四旋翼无人机的独轮车模型。

Since the last layer of the actor network is activated by a hyperbolic tangent function, we resize and normalized the action space in order to fully utilize the numeric range of the action space.

由于演员网络最后一层由双曲正切函数激活，我们调整并归一化动作空间，以充分利用动作空间的数值范围。

C. Reward Function Design

C. 奖励函数设计

We adopt the design of reward function in [11] and added an addition penalty value which is described in the next paragraph, and the algorithm is shown in Algorithm 1.

我们采用了文献 [11] 中的奖励函数设计，并增加了一个额外的惩罚值，该惩罚值在下一段中有所描述，算法如算法 1 所示。

Algorithm 1 REWARD FUNCTION $\mathcal{R}$

算法 1 奖励函数 $\mathcal{R}$

```
Input: $d_{\text{goal},t} d_{\text{goal},t+1} \theta_{\text{goal},t+1} d_{\text{obs},t+1} d_{\text{init}} v_{\max} d_{\text{colli}} d_{\min}$ Output: $r_t$
输入: $d_{\text{goal},t} d_{\text{goal},t+1} \theta_{\text{goal},t+1} d_{\text{obs},t+1} d_{\text{init}} v_{\max} d_{\text{colli}} d_{\min}$ 输出: $r_t$
if $d_{obs,t+1} \leq d_{colli}$ then
如果 $d_{obs,t+1} \leq d_{colli}$ 那么
    $r_t \leftarrow -2$
else
else
    if $r_t > 0$ then
    如果 $r_t > 0$ 那么
        $r_t \leftarrow r_t \left(1 - \frac{d_{\text{goal},t+1}}{1.5 d_{\text{init}}}\right)$
    else
    else
        $r_t \leftarrow r_t \left(1 + \frac{d_{\text{goal},t+1}}{1.5 d_{\text{init}}}\right)$
    end if
    结束 if
    $r_t \leftarrow r_t - 0.01 \left|\theta_{\text{goal},t+1}\right|$
    $r_t \leftarrow r_t - 0.01 \min\left(\frac{v_{\max}}{d_{\text{init}}}, 1\right)$
end if
结束 if
```

Let $\mathbf{r}_t$ denotes the reward agent receives at time $t \in [0, \infty)$ and $\mathcal{R}: \mathcal{S} \rightarrow \mathbf{r}_t$ be the reward function where $\mathcal{S}$ is the global state of the environment. Let d_{colli} be the collision threshold for reward calculation and $d_{obs,t+1}$ be the relative obstacle distance of the new state $\mathbf{s}_{t+1}$. If $d_{obs,t+1} < d_{\text{colli}}$, the agent is punished by a reward value $\mathrm{r}_t = -2$ to help it learn how to avoid collisions. Otherwise, r_t is initialized with a value proportional to the relative velocity between the agent and its goal. The reward is positive as the agent moves towards its goal and negative as the agent moves away from its goal.

令 $\mathbf{r}_t$ 表示代理在时间 $t \in [0, \infty)$ 收到的奖励，$\mathcal{R}: \mathcal{S} \rightarrow \mathbf{r}_t$ 是奖励函数，其中 $\mathcal{S}$ 是环境的全局状态。令 d_{colli} 为计算奖励的碰撞阈值，$d_{obs,t+1}$ 为新状态 $\mathbf{s}_{t+1}$ 的相对障碍物距离。如果 $d_{obs,t+1} < d_{\text{colli}}$ ，代理将受到一个奖励值 $\mathrm{r}_t = -2$ 的惩罚，以帮助它学习如何避免碰撞。否则，r_t 初始化为与代理和目标之间的相对速度成比例的值。当代理向目标移动时，奖励为正，当代理远离目标时，奖励为负。

The initial reward value is also scaled by a factor to make the reward value larger as the agent approaches the target even if the speed of the agent remains unchanged. $|\theta_{goal,t+1}|$ is then subtracted from $\mathbf{r}_t$ after multiplied with a small factor. This helps the agent learns to navigate to the goal by applying punishments when it is not flying in the direction of the goal. Finally, an addition value inversely proportional to the initial goal distance d_{init} is subtract from $\mathbf{r}_t$ to encourage the agent to learn to reach its target as soon as possible.

初始奖励值也通过一个因子进行缩放，使得即使代理的速度保持不变，当代理接近目标时，奖励值也会变大。$|\theta_{goal,t+1}|$ 然后从 $\mathbf{r}_t$ 中减去，之前已乘以一个小的因子。这有助于代理在不是飞向目标方向时

通过施加惩罚来学习导航至目标。最后，一个与初始目标距离 d_{init} 成反比的附加值从 $\mathbf{r}_t$ 中减去，以鼓励代理尽快学会到达其目标。

D. Soft Actor Critic Framework in UAV Collision Avoidance

D. 无人机避障中的软行为评价器框架

Conventional reinforcement learning aims to learn a policy $\pi\left(\mathbf{a}_t, \mathbf{s}_t\right)$ that maximize the expected sum of rewards $\sum_t \mathbb{E}_{\left(\mathbf{s}_t, \mathbf{a}_t\right) \sim \rho_\pi}\left[r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$ where ρ_π denotes the state action distribution of policy π and $\mathbb{E}$ denotes the expectation value function. However, SAC generalizes the standard objective by augmenting it with an entropy term $\mathcal{H}(P)=\mathbb{E}_{x \sim P}[-\log P(x)]$ where x is a random variable with probability density function P with respect to x , such that the optimal policy π^* additionally aims to maximize its entropy at each visited state [28]:

传统强化学习旨在学习一个策略 $\pi\left(\mathbf{a}_t, \mathbf{s}_t\right)$ 以最大化期望奖励之和 $\sum_t \mathbb{E}_{\left(\mathbf{s}_t, \mathbf{a}_t\right) \sim \rho_\pi}\left[r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$ ，其中 ρ_π 表示策略的状态动作分布 π ，$\mathbb{E}$ 表示期望值函数。然而，SAC 通过增加一个熵项 $\mathcal{H}(P)=\mathbb{E}_{x \sim P}[-\log P(x)]$ 来推广标准目标，其中 x 是一个随机变量，其概率密度函数 P 关于 x ，使得最优策略 π^* 还旨在在访问的每个状态处最大化其熵 [28]:

$$\pi^*=\arg \max_\pi \sum_t \mathbb{E}_{\left(\mathbf{s}_t, \mathbf{a}_t\right) \sim \rho_\pi}\left[r\left(\mathbf{s}_t, \mathbf{a}_t\right)+\alpha \mathcal{H}\left(\pi\left(\cdot \mid \mathbf{s}_t\right)\right)\right] \tag{7}$$

where $\alpha>0$ is the temperature parameter that determines the relative importance of the entropy term versus the reward $r\left(\mathbf{s}_t, \mathbf{a}_t\right)$. The concept of the implemented entropy term $\mathcal{H}\left(\pi\left(\cdot \mid \mathbf{s}_t\right)\right)$ represents the randomness of the outputs of the stochastic policy π with given $\mathbf{s}_t$. The extra entropy term incentivizes the policy to explore more widely, improving its robustness against perturbations [29].

其中 $\alpha>0$ 是温度参数，它决定了熵项相对于奖励 $r\left(\mathbf{s}_t, \mathbf{a}_t\right)$ 的相对重要性。实现的熵项 $\mathcal{H}\left(\pi\left(\cdot \mid \mathbf{s}_t\right)\right)$ 的概念表示了给定 $\mathbf{s}_t$ 的随机策略 π 输出的随机性。额外的熵项激励策略更广泛地探索，提高其对抗扰动的鲁棒性 [29]。

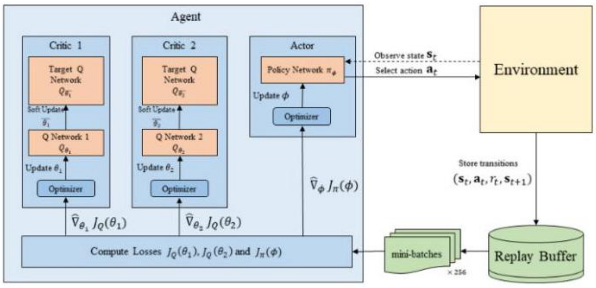
Agent
Critic 1
Target Q Network $Q_{\bar{\theta}_1}$
Soft Update
$\bar{\theta}_1$
Q Network 1 Q_{θ_1}
Update θ_1
Optimizer
Critic 2
Target Q Network $Q_{\bar{\theta}_2}$
Soft Update
$\bar{\theta}_2$
Q Network 2 Q_{θ_2}
Update θ_2
Optimizer
Actor
Policy Network π_ϕ
Update ϕ
Optimizer
Observe state $\mathbf{s}_t$
Select action $\mathbf{a}_t$
Environment
Store transitions
$(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$
$\hat{\nabla}_\phi J_\pi(\phi)$
$\hat{\nabla}_{\theta_1} J_Q(\theta_1)$
$\hat{\nabla}_{\theta_2} J_Q(\theta_2)$
Compute Losses $J_Q(\theta_1), J_Q(\theta_2)$ and $J_\pi(\phi)$
mini-batches
× 256
Replay Buffer

Fig. 3. An illustration of a Soft Actor Critic framework.

图 3. Soft Actor Critic 框架的示意图。

The SAC deep reinforcement learning algorithm framework is shown in Fig. 3. It is composed of five neural networks: A Gaussian policy network π_ϕ with parameters ϕ , two Q-networks Q_{θ_1} and Q_{θ_2} with parameters θ_1 and θ_2 , and two target Q-networks $Q_{\bar{\theta}_1}$ and $Q_{\bar{\theta}_2}$ with parameters $\bar{\theta}_1$ and $\bar{\theta}_2$. In order to train the deep reinforcement learning model, we must first compute the loss function of each network in the SAC framework. As suggested in [28], we learn the Q-network parameters as a regression problem by minimizing the following loss function:

图 3 展示了 SAC 深　姆强化学习算法框架。它由五个神经网络组成: 一个具有参数 π_ϕ 的 Gaussian 策略网络 ϕ ，两个具有参数 θ_1 和 θ_2 的 Q 网络 Q_{θ_1} 和 Q_{θ_2} ，以及两个具有参数 $\bar{\theta}_1$ 和 $\bar{\theta}_2$ 的目标 Q 网络 $Q_{\bar{\theta}_1}$ 和 $Q_{\bar{\theta}_2}$ 。为了训练深度强化学习模型，我们首先必须计算 SAC 框架中每个网络的损失函数。如 [28] 所建议，我们通过最小化以下损失函数来学习 Q 网络参数作为一个回归问题:

$$J_Q\left(\theta_i\right)=\mathbb{E}_{\left(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}\right) \sim \mathcal{D}}\left[\left(Q_{\theta_i}\left(\mathbf{s}_t, \mathbf{a}_t\right)-\left(r\left(\mathbf{s}_t, \mathbf{a}_t\right)+\gamma V_{\bar{\theta}_1, \bar{\theta}_2}\left(\mathbf{s}_{t+1}\right)\right)\right)^2\right] \tag{8}$$

using mini-batches from the replay buffer $\mathcal{D}$, where the value function $V_{\bar{\theta}_1, \bar{\theta}_2}$ is implicitly defined through the Q-networks and the policy as:

使用来自重放缓冲区 $\mathcal{D}$ 的迷你批次，其中值函数 $V_{\bar{\theta}_1, \bar{\theta}_2}$ 通过 Q 网络和策略隐式定义如下:

$$V_{\bar{\theta}_1, \bar{\theta}_2}\left(\mathbf{s}_t\right)=\mathbb{E}_{\mathbf{a}_t \sim \pi}\left[\min _{i \in\{1,2\}} Q_{\bar{\theta}_i}\left(\mathbf{s}_t, \mathbf{a}_t\right)-\alpha \log \pi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)\right] \tag{9}$$

Then, we improve the Gaussian policy in a similar factor by minimizing:
然后，我们通过最小化以下因素来改进高斯策略:

$$J_\pi(\phi)=\mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}, \mathbf{a}_t \sim \pi}\left[\alpha \log \pi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)-\min _{i \in\{1,2\}} Q_{\theta_i}\left(\mathbf{s}_t, \mathbf{a}_t\right)\right] \tag{10}$$

using the reparameterization trick. We re-parameterize the policy function using a neural network transformation of $\mathbf{a}_t=f_\phi\left(\varepsilon_t \mid \mathbf{s}_t\right)$ where ε_t is an input noise vector sampled from a normal Gaussian distribution as suggested in [28]. Furthermore in [28], the author proposed a way to automate the process of choosing the optimal temperature α . Instead of requiring the user to set the temperature manually, they automate this process by formulating a different maximum entropy reinforcement learning objective:

使用重参数化技巧。我们使用神经网络转换重参数化策略函数，该转换采用 $\mathbf{a}_t=f_\phi\left(\varepsilon_t \mid \mathbf{s}_t\right)$ 作为输入噪声向量，该向量是从标准高斯分布中采样的，如文献 [28] 中所建议。此外，在文献 [28] 中，作者提出了一种自动化选择最优温度 α 的方法。他们通过构建一个不同的最大熵强化学习目标函数来自动化这个过程，而不是要求用户手动设置温度:

$$J(\alpha)=\mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}, \mathbf{a}_t \sim \pi_\phi}\left[-\alpha \log \pi_\phi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)-\alpha \overline{\mathcal{H}}\right] \tag{11}$$

where $\overline{\mathcal{H}}$ denote the target entropy and is set equal to the dimension of the action space. As in [28] we select target entropy as the dimension of the action space, letting $\overline{\mathcal{H}}=-\operatorname{dim}\left(\mathbf{a}_t\right)=-2$. Finally, the target Q-networks $Q_{\bar{\theta}_i}, i \in\{1,2\}$ are updated with a delay factor τ with respect to the original Q-networks as shown in (12).

其中 $\overline{\mathcal{H}}$ 表示目标熵，设置为动作空间的维度。与文献 [28] 中的做法一样，我们选择动作空间的维度作为目标熵，让 $\overline{\mathcal{H}}=-\operatorname{dim}\left(\mathbf{a}_t\right)=-2$ 。最后，目标 Q 网络 $Q_{\bar{\theta}_i}, i \in\{1,2\}$ 根据延迟因子 τ 相对于原始 Q 网络进行更新，如公式 (12) 所示。

$$\bar{\theta}_i \leftarrow \tau \theta_i+(1-\tau) \bar{\theta}_i, i \in\{1,2\} \tag{12}$$

Algorithm 2 ALGORITHM OF SOFT ACTOR-CRITIC WITH AUTOMATIC ENTROPY ADJUSTMENT

算法 2 自动调整熵的 SOFT ACTOR-CRITIC 算法

```
Initialize network parameters $\theta_1, \theta_2, \bar{\theta}_1, \bar{\theta}_2, \phi$ and entropy temperature coefficient $\alpha$
初始化网络参数 $\theta_1, \theta_2, \bar{\theta}_1, \bar{\theta}_2, \phi$ 和熵温度系数 $\alpha$
foreach episode do
对每个剧集进行迭代
    foreach environment step do
    对每个环境步骤进行迭代
        foreach angent do
        对每个智能体进行迭代
            $\mathbf{a}_t \sim \pi_\phi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)$
            $\mathbf{s}_{t+1}, r_t \sim$ environment $\left(\mathbf{s}_{t+1}, r_t \mid \mathbf{s}_t, \mathbf{a}_t\right)$
            $\mathbf{s}_{t+1}, r_t \sim$ 环境 $\left(\mathbf{s}_{t+1}, r_t \mid \mathbf{s}_t, \mathbf{a}_t\right)$
            $\mathcal{D} \leftarrow \mathcal{D} \cup\left\{\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}\right\}$
        end
        结束
        $\theta_i \leftarrow \theta_i-\lambda \widehat{\nabla}_\theta J_Q\left(\theta_i\right)$ for $i \in\{1,2\}$
        $\theta_i \leftarrow \theta_i-\lambda \widehat{\nabla}_\theta J_Q\left(\theta_i\right)$ 对于 $i \in\{1,2\}$
        $\phi \leftarrow \phi-\lambda \widehat{\nabla}_\phi J_\pi(\phi)$
        $\alpha \leftarrow \alpha-\lambda \widehat{\nabla}_\alpha J_\pi(\alpha)$
        $\bar{\theta}_i \leftarrow \tau \theta_i+(1-\tau) \bar{\theta}_i$ for $i \in\{1,2\}$
        $\bar{\theta}_i \leftarrow \tau \theta_i+(1-\tau) \bar{\theta}_i$ 用于 $i \in\{1,2\}$
    end
    结束
end
结束
```

While training, all objectives in Equations (8), (10), and (11) are all optimized simultaneously. Algorithm 2 summarizes the full training procedure, where $\widehat{\nabla}$ denotes stochastic gradients and λ denotes the learning rate. The parameters are first initialized. During environment steps of each iteration, the algorithm will sample action, $\mathbf{a}_t$, from action space, $\pi_{\varphi}\left(\mathbf{a}_t \mid \mathbf{s}_t\right)$, and transition state, $\mathbf{s}_{s+1}$, from the environment regarding current environment state, $\mathbf{s}_t$ and action taken, $\mathbf{a}_t$. These data with the reward $\mathbf{r}_t$, will then be stored in the replay buffer, $\mathcal{D}$ for each agent. The parameters then will be optimized using gradient descend on the basis of equation (8), (10), (11) and

在训练过程中，方程 (8)、(10) 和 (11) 中的所有目标都是同时优化的。算法 2 总结了完整的训练过程，其中 $\widehat{\nabla}$ 表示随机梯度，λ 表示学习率。参数首先初始化。在每次迭代的环境步骤中，算法将从动作空间 $\pi_{\varphi}\left(\mathbf{a}_t \mid \mathbf{s}_t\right)$ 中采样动作 $\mathbf{a}_t$ ，并从当前环境状态 $\mathbf{s}_t$ 和采取的动作 $\mathbf{a}_t$ 相关的环境中转换状态 $\mathbf{s}_{s+1}$ 。这些数据以及奖励 $\mathbf{r}_t$ 将被存储在每个智能体的回放缓冲区 $\mathcal{D}$ 中。然后，将基于方程 (8)、(10)、(11) 使用梯度下降法来优化参数，(12).

E. Training Environment

E. 训练环境

To increase training efficiency, instead of using a physics engine, we create a custom OpenAI Gym [30] environment for developing and testing learning agents written in Python. In each episode, the environment is initialized by generating M agents with randomly distributed depots $\mathcal{D}=\{D_1, D_2, \ldots, D_m\}$ and goals $\mathcal{G}=\{G_1, G_2, \ldots, G_m\}$, where $\forall i, j \in \mathcal{M}$ and i $\neq$ j $G_i \neq G_j$. At each time slot t the environment takes m sets of actions $\mathbf{a}_t$ and returns m sets of new states $\mathbf{s}_{t+1}$ and rewards $\mathbf{r}_t$ with respect to each agent m . The goal of the environment is to let the model learn a single policy over all agents while all agents can successfully reach their targets and avoid collisions. Each episode of training is terminated either when the agent reaches its goal or when the agent flies out of the boundary. Each episode will end only when all agents have reached their goals or time has run out.

为了提高训练效率，我们不是使用物理引擎，而是创建了一个定制的 OpenAI Gym [30] 环境，用于开发和测试用 Python 编写的学习智能体。在每一集中，环境通过生成具有随机分布的仓库 $\mathcal{D}=\{D_1, D_2, \ldots, D_m\}$ 和目标 $\mathcal{G}=\{G_1, G_2, \ldots, G_m\}$ 的 M 个智能体来初始化，其中 $\forall i, j \in \mathcal{M}$ 和 i $\neq$ j $G_i \neq G_j$ 。在每一个时间槽 t 中，环境采取 m 组动作 $\mathbf{a}_t$ ，并返回 m 组新状态 $\mathbf{s}_{t+1}$ 和每个智能体 m 的奖励 $\mathbf{r}_t$ 。环境的目标是让模型学习所有智能体的单一策略，同时所有智能体都能成功到达目标并避免碰撞。每次训练的集数在智能体达到目标或飞出边界时终止。只有在所有智能体都达到目标或时间耗尽时，每个集数才会结束。

V. ROUTE OPTIMIZATION IN UAV PARCEL DELIVERY TASKS

V. 无人机包裹配送任务中的路线优化

A. Modified CVRP

A. 修改后的 CVRP

We discuss the solution to the route optimization in UAV delivery system in this section. we will formulate the modified CVRP (mCVRP) with the proposed energy cost function in (2), which is consider an extension of the conventional CVRP

我们在本节讨论无人机配送系统中路由优化的解决方案。我们将使用提出的能量成本函数 (2) 来构建修改后的 CVRP(mCVRP)，这可以看作是传统 CVRP 的扩展。

problem.

问题。

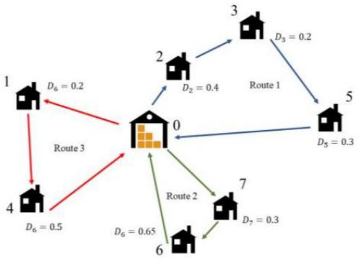

Fig. 4. Example of a solution of the mCVRP.

图 4. mCVRP 解决方案的示例。

Consider an environment with a total number of n nodes and $n-1$ customers as illustrated in Fig. 4., we denote the location of each node as L_i where $i \in \{1,2,\ldots,n\}$.L_0 is defined as the location of the depot while $L_{i,i\neq 0}$ is the location of the customers, the distance between node i and node j is represented by $d_{i,j}$. It is assumed that each UAV is able to carry at most C_p parcels at once and have a payload limit of C_w . The solution space of the proposed mCVRP is composed of three variables: $x_{i,j}, w_{i,j}$ and $p_{i,j}, x_{i,j}$, is a binary number that is equal to 1 if a UAV goes from node i to node j.$w_{i,j}$ is the total weight of the parcels that the UAV is carrying when going from node i to node j.$p_{i,j}$ is the number of parcels that a UAV is carrying when flying from node i to node j . With the presented parameters and the variables, the mathematical formulation of mCVRP is the following:

考虑一个由 n 个节点和 $n-1$ 个客户组成的环境，如图 4 所示。我们用 L_i 表示每个节点的位置，其中 $i \in \{1,2,\ldots,n\}$.L_0 被定义为仓库的位置，而 $L_{i,i\neq 0}$ 是客户的位置，节点 i 和节点 j 之间的距离由 $d_{i,j}$ 表示。假设每架无人机一次最多能携带 C_p 个包裹，并且有 C_w 的载重限制。所提出 mCVRP 的解空间由三个变量组成: $x_{i,j}, w_{i,j}$ 和 $p_{i,j}, x_{i,j}$ ，是一个二进制数，如果无人机从节点 i 飞往节点 j.$w_{i,j}$ ，则等于 1； j.$p_{i,j}$ 是无人机从节点 i 飞往节点 j 时携带的包裹总重量；[latex14] 是无人机从节点 i 飞往节点 j 时携带的包裹数量。根据所提供的参数和变量，mCVRP 的数学表述如下:

Objective:

目标:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_{i,j} d_{i,j} \left(1 + w_{i,j}\right) \tag{13}$$

Subject to:

约束条件:

$$\sum_{j=1}^{n} x_{i,j} = 1 \ \forall i = \{2,\ldots,n\} \tag{14}$$

$$\sum_{j=1}^{n} x_{j,i} = 1 \ \forall i = \{2,\ldots,n\} \tag{15}$$

$$\sum_{j=1}^{n} \left(w_{j,i} - w_{i,j}\right) = D_i \ \forall i = \{2,\ldots,n\} \tag{16}$$

$$\sum_{j=1}^{n} \left(p_{j,i} - p_{i,j}\right) = 1 \ \forall i = \{2,\ldots,n\} \tag{17}$$

$$0 \leq w_{i,j} \leq C_w x_{i,j} \ \forall i,j = \{1,\ldots,n\} \tag{18}$$

$$0 \leq p_{i,j} \leq C_p x_{i,j} \ \forall i,j = \{1,\ldots,n\} \tag{19}$$

$$x_{i,i} = 0 \ \forall i = \{1,\ldots,n\} \tag{20}$$

$$x_{i,j} \in \{0,1\} \ \forall i,j = \{1,\ldots,n\} \tag{21}$$

The objective function (13) aims to minimize the total travel energy cost of all UAVs. Equation (14) and (15) ensure that only one UAV enters or leaves the node i except for the depot. Equation (16) constrains the variable $w_{i,j}$ according to the customer demand D_i . Equation (17) limits the variable $p_{j,i}$ since each customer only demands one package. Equation (18) and (19) are the key constraints that limit the maximum weight and the number of payloads a UAV can carry at each route. Equation (20) makes sure no nodes will be skipped. Equation (21) limits the forces $x_{i,j}$ become a Boolean value.

目标函数 (13) 旨在最小化所有无人机的总旅行能耗成本。方程 (14) 和 (15) 确保除仓库外，只有一个无人机进入或离开节点 i 。方程 (16) 根据客户需求 D_i 约束变量 $w_{i,j}$ 。方程 (17) 限制变量 $p_{j,i}$ ，因为每个客户仅需求一个包裹。方程 (18) 和 (19) 是关键约束，限制了无人机在每条路线上可以携带的最大重量和载荷数量。方程 (20) 确保没有节点会被跳过。方程 (21) 限制力 $x_{i,j}$ 变为布尔值。

B. Adopting Genetic Algorithm

B. 采用遗传算法

We now adopt genetic algorithm to solve the proposed nonlinear programming problem. We would state the approach encoding a set of solutions into chromosome and the design of fitness function in order to implement genetic algorithm in this section. The solution to an mCVRP is sets as: $\{\{0 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 0\}, \{0 \rightarrow 7 \rightarrow 6 \rightarrow 0\}, \{0 \rightarrow 1 \rightarrow 4 \rightarrow 0\}\}$ As shown in Fig. 4. Each element in the sets indicates the ID of the customer and ID 0 is the depot. We can directly encode the sequence of the solution set by eliminating the depots as Fig. 5. Repeat pattern is not allowed since each customer only receives one parcel in each mission. Each segment of the chromosome is determined by the summation of demands, once $\sum D_i$ is greater than the UAV payload limit C_w , another UAV is needed to deliver the extra parcel. The length of the chromosome is a constant equal to $n-1$. It is worth noticing that different combination with identical fitness values is possible. For example, Fig. 5 shows 5 redundant results having the exact routing result.

我们现在采用遗传算法来解决所提出的非线性规划问题。我们将陈述将解决方案集编码为染色体以及设计适应度函数的方法，以在本节中实现遗传算法。mCVRP 的解决方案被设置为: $\{\{0 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 0\}, \{0 \rightarrow 7 \rightarrow 6 \rightarrow 0\}, \{0 \rightarrow 1 \rightarrow 4 \rightarrow 0\}\}$ 如图 4 所示。集合中的每个元素表示客户的 ID，ID 0 是仓库。我们可以通过消除仓库直接编码解决方案集的序列，如图 5 所示。重复模式是不允许的，因为每次任务中每个客户只接收一个包裹。染色体的每个片段由需求的累加确定，一旦 $\sum D_i$ 大于无人机的载荷限制 C_w ，就需要另一个无人机来递送额外的包裹。染色体的长度是一个常数，等于 $n-1$ 。值得注意的是，具有相同适应度值的不同组合是可能的。例如，图 5 显示了 5 个冗余结果，它们具有完全相同的路由结果。

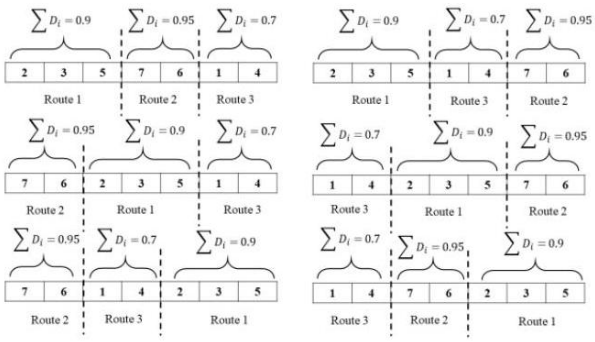

Fig.5. Example of chromosome encoding.

图 5. 染色体编码示例。

We establish a function to evaluate the fitness of a chromosome, which is set to be the reciprocal of total energy. The energy could be calculated via (2). We implement a 3-way tournament selection as our selection strategy, selecting 3 individuals randomly and running a tournament among them. Only the fittest candidate is chosen and passed to perform crossover and mutation processes. The details of crossover and mutation operations are described below. The elitism strategy is also adopted to ensure that the fittest solution is always retained without undergoing crossover or mutation operations.

我们建立一个函数来评估染色体的适应性，该函数被设置为总能量的倒数。能量可以通过公式 (2) 计算。我们实施了一种 3-way 锦标赛选择策略，随机选择 3 个个体并在它们之间进行锦标赛。只选择最适应的候选者进行交叉和变异过程。下面描述了交叉和变异操作的细节。我们还采用了精英策略，以确保最适应的解决方案在不经历交叉或变异操作的情况下始终被保留。

1) Crossover Operation: We now adopt Order Crossover genetic algorithm to solve the proposed nonlinear programming problem. To implement Order Crossover (OX), we need 2 chromosomes, denoted as parent 1 and parent 2, in advance. The steps are as follows:

1) 交叉操作: 我们现在采用顺序交叉遗传算法来解决提出的非线性规划问题。为了实现顺序交叉 (OX)，我们需要提前准备 2 个染色体，分别称为父代 1 和父代 2。步骤如下:

Step (1) Select a random segment from parent 1.

步骤 (1) 从父代 1 中随机选择一个片段。

Step (2) Place selected segment at the corresponding position of a new offspring.

步骤 (2) 将选定的片段放置到新后代中相应的位置。

Step (3) Remove the element existing in the selected segment from parent 2.

步骤 (3) 从父代 2 中删除存在于选定片段中的元素。

Step (4) Place the rest of part of parent 2 at the unfixed position of offspring 1 from left to right according to the order of the sequence.

步骤 (4) 将父代 2 的其余部分从左到右根据序列的顺序放置到后代 1 的未固定位置。

Step (5) Go back to Step (1) and swap parents to generate another offspring.

步骤 (5) 回到步骤 (1) 并交换父代以生成另一个后代。

2) Mutation Operation: The mutation operation is performed by arbitrarily selecting a segment of random length in a chromosome and flipping the order of the selected segment. This kind of mutation operation will not break the integrity of the chromosome.

2) 变异操作: 变异操作通过在染色体中任意选择一个随机长度的片段并翻转选定片段的顺序来执行。这种变异操作不会破坏染色体的完整性。

VI. TEST RESULTS

VI. 测试结果

A. DRL-based Method Training Results

A. 基于 DRL 的方法训练结果

The adopted method is trained in a square map with a width of 50 meters and a total of 5 agents. The maximum velocity $V_{\max}$ of each agent is 10 m/s and the maximum acceleration is $5 \mathrm{~m/s}^2$. The sensing range d_{sense} the collider radius d_{colli} of each UAV is set to 15 m and 1 m respectively. The agent interacts with the environment 50 times per second, i.e., the agent takes 50 actions in a second. The parameters of the SAC method are shown in Table I.

采用的方法在一个宽 50 米的方形地图中进行训练，共有 5 个代理。每个代理的最大速度 $V_{\max}$ 是 10 m/s ，最大加速度是 $5 \mathrm{~m/s}^2$ 。每个无人机的感知范围 d_{sense} 和碰撞半径 d_{colli} 分别设置为 15 m 和 1 m 。代理每秒与环境互动 50 次，即代理每秒采取 50 个动作。SAC 方法的参数如表 I 所示。

TABLE I. SOFT ACTOR CRITIC HYPERPARAMETERS	
Parameters	Methods/Values
Optimizer	Adam (Kingma, Jimmy Ba (2015))
Learning Rate	3 \times 10(̃-4)
Reward Discount	0.99
Replay Buffer Size	106
Number of hidden layers	2
Number of hidden units per layer	256
Number of samples per minibatch	256
Target Entropy	-2
Activation Function	ReLU
Target Smoothing Coefficient	0.005
Temperature Coefficient	0.5

表 I. SOFT ACTOR CRITIC 超参数	
参数	方法/值
优化器	Adam (Kingma, Jimmy Ba (2015))
学习率	3 × 10(̃-4)
奖励折扣	0.99
重放缓冲区大小	106
隐藏层的数量	2
每层的隐藏单元数量	256
每个迷你批次的样本数量	256
目标熵	-2
激活函数	ReLU
目标平滑系数	0.005
温度系数	0.5

In Fig. 6, we illustrate the learning curve with respect to the episode score of the adopted SAC method. Since the goal of the agent is randomly generated and the total reward is positively correlated with the initial target distance, the score oscillates even if the agent successfully reaches the target. The huge spikes in the unsmoothed curve result from agents failing to avoid collisions and getting tangled with others.

在图 6 中，我们展示了采用 SAC 方法的得分随剧集分数的学习曲线。由于代理的目标是随机生成的，且总奖励与初始目标距离成正比，即使代理成功到达目标，得分也会波动。未平滑曲线中的巨大尖峰是由于代理未能避免碰撞并与其他代理纠缠在一起造成的。

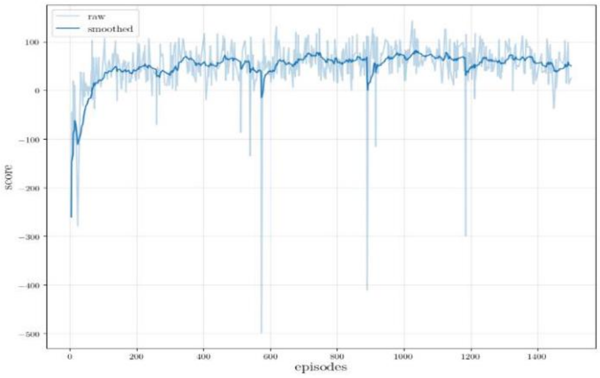

Fig. 6. Training curve of the adopted reinforcement learning algorithm.

图 6。采用强化学习算法的训练曲线。

In Figs. 7 and 8, we compare the training curves of different deep reinforcement learning methods with respect to the success rate(SR)and the collision rate (CR) .SR is defined as the number of agents that successfully reaches their target divided by the number of total agents in the simulation. CR is defined as the number of agents collides with other agents divided by the number of total agents in the environment. The result is produced by running 10 episodes of evaluation every 10 episodes while training. The training result shows that the success rate of the adopted SAC method outperforms other famous deep reinforcement methods such as DDPG and TD3. Not only does the adopted method converge faster, but it also performs relatively stable compared to others. Although DDPG seems to learn faster at the beginning, it fails to learn the collision avoidance policy, causing a high collision rate. As for TD3, the result shows that it may be able to learn a good policy concerning the collision rate, but the success rate still does not converge after 1500 episodes of training. In contrast, SAC converges faster and tends to have a more stable performance after 1000 episodes of training. Compared to the Q-learning method proposed in [11], the SAC method provides not only a higher success rate but also continuous UAV motion control instead of discrete UAV heading control. The overall success rate of the validation result is also higher than the Q-learning method with the same number of agents. different deep reinforcement learning algorithms including SAC (blue), DDPG (green) and TD3 (orange). SAC (blue), DDPG (green) and TD3 (orange).

在图 7 和图 8 中，我们比较了不同深度强化学习方法的训练曲线，这些曲线与成功率 (SR) 和碰撞率 (CR) .SR 相关，碰撞率定义为成功到达目标的代理数量除以模拟中总代理数量的比值。CR 定义为与环境中的其他代理发生碰撞的代理数量除以总代理数量的比值。该结果是通过在训练过程中每 10 个周期运行 10 个评估周期得到的。训练结果显示，采用 SAC 方法的成功率超过了其他著名的深度强化学习方法，如 DDPG 和 TD3。不仅该方法收敛速度更快，而且与其他方法相比，表现相对稳定。尽管 DDPG 在开始时似乎学习得更快，但它无法学习避障策略，导致碰撞率较高。至于 TD3，结果显示它可能能够学习到关于碰撞率的良好策略，但在 1500 个训练周期后，成功率仍然没有收敛。相比之下，SAC 收敛速度更快，在 1000 个训练周期后表现更加稳定。与文献 [11] 中提出的 Q 学习方法相比，SAC 方法不仅提供了更高的成功率，而且还能提供连续的无人机运动控制，而不是离散的无人机航向控制。在相同数量的代理下，验证结果的总体成功率也高于 Q 学习方法。包括 SAC(蓝色)、DDPG(绿色) 和 TD3(橙色) 的不同深度强化学习算法。

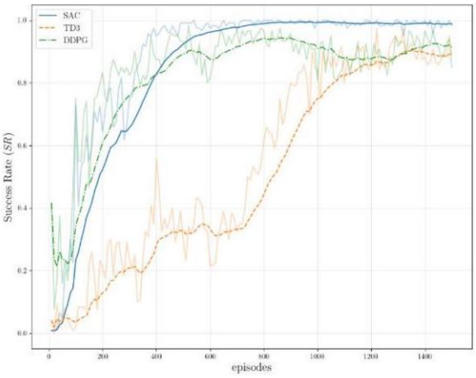

Fig. 7. Success rate with respect to trained episodes of
图 7. 成功率与训练周期数的关系

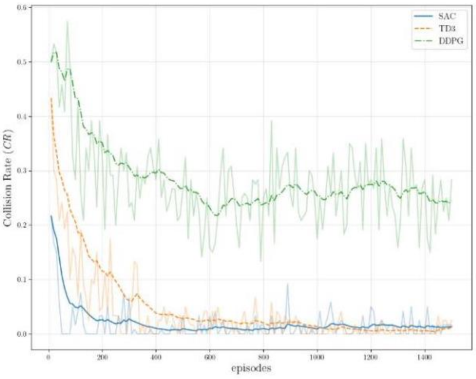

Fig. 8. Collision rate with respect to trained episodes of different deep reinforcement learning algorithms including
图 8. 碰撞率与不同深度强化学习算法训练周期数的关系

To further examine the performance of the trained SAC method, we tested the learned policy under environments with different numbers of total agents. Each number of total agents is tested with 100 random episodes. In Fig. 9, it shows that as the total agent number increases, the success rate starts descending, and the collision rate starts ascending. As the environment gets more crowded, the hard limitation of the input state that only allows the agent to perceive at most two nearby obstacles at once becomes more significant. Note that the maximum step of each validation episode is 1500 steps. It is considered failed if an agent is unable to reach its goal within the step limit. The result in Fig. 9 also illustrates the extensibility of the adopted SAC method. The trained policy still has a collision rate below 1 percent and a success rate above 99.9 percent when the total agent number is 12, although the model is trained in an environment with only 5 agents.

为了进一步检验训练后的 SAC 方法的性能，我们在具有不同总数代理的环境下测试了学习到的策略。每个总代理数都进行了 100 次随机回合的测试。在图 9 中，它显示随着总代理数的增加，成功率开始下降，碰撞率开始上升。随着环境变得更加拥挤，输入状态的硬限制，即只允许代理同时感知最多两个附近障碍物的限制变得更加显著。注意，每个验证回合的最大步数是 1500 步。如果代理在步数限制内无法达到目标，则视为失败。图 9 的结果还说明了所采用 SAC 方法的扩展性。即使模型是在只有 5 个代理的环境中训练的，当总代理数为 12 时，训练出的策略仍然能够保持低于 1% 的碰撞率和高于 99.9% 的成功率。

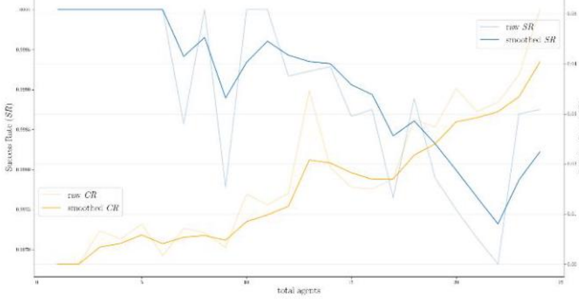

Fig. 9. Success rate(blue) and collision rate(orange) with respect to different numbers of agents in the environment. Light color indicates raw data, while dark color indicates smoothed data.

图 9. 环境中不同数量代理的成功率 (蓝色) 和碰撞率 (橙色)。浅色表示原始数据，深色表示平滑后的数据。

Finally, we illustrate the result of implementing the proposed DRL-based UAV collision avoidance method with the trained model in a realistic simulation environment using ROS and Gazebo. The maximum velocity $V_{\max}$ of each UAV agent is 10 m/s and the maximum acceleration is 5 m/s^2 . The sensing range d_sense of the UAVs is set to 15 m . The trajectory of four UAV agents trying to avoid collisions between each other using the proposed method while swapping their positions is shown in Fig. 10. The result shows that even if the proposed deep reinforcement learning based model is trained in a simplified simulated environment, it can still be adopted in a realistic physics engine with sensor noises and control delays. B. Test Results of the Genetic Algorithm Based Route Optimization in Simulated Parcel Delivery System

最后，我们展示了在真实的模拟环境中使用 ROS 和 Gazebo 实施所提出基于深度强化学习的无人机避障方法，并使用训练模型的成果。每个无人机代理的最大速度 $V_{\max}$ 是 10 m/s ，最大加速度是 5 m/s^2 。无人机的感知范围 d_sense 设置为 15 m 。图 10 显示了四个无人机代理在尝试使用所提出的方法避免相互碰撞并在交换位置时的轨迹。结果表明，即使提出的基于深度强化学习的模型是在简化的模拟环境中训练的，它仍然可以被应用于具有传感器噪音和控制延迟的真实物理引擎中。B. 基于遗传算法的路线优化在模拟包裹递送系统中的测试结果。

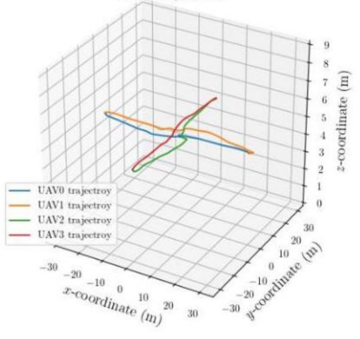

Fig. 10. Trajectory of 4 UAVs trying to avoid each other.

图 10. 4 架无人机试图避免相互碰撞的轨迹。

The genetic algorithm converges after about 1500 generations with the parameters shown in Table II. It took 8.3 seconds to evolve 3000 generations on a laptop with Intel® Core $^{\mathrm{TM}}$ i7-12650H CPU. The optimized result leads the UAV to first deliver the heaviest and nearest parcel in order to reduce its overall energy consumption. The routing result of the proposed genetic algorithm in the scenario of 20 customers is shown in Fig. 11. Each blue dot in the figure represents a customer and the number beside it denotes the demand D_i and the center red dot represents the depot.

遗传算法在大约 1500 代后收敛，参数如表 II 所示。在一台配备 Intel® Core $^{\mathrm{TM}}$ i7-12650H CPU 的笔记本电脑上，进化 3000 代耗时 8.3 秒。优化结果使得无人机首先递送最重且最近的包裹，以减少其总体能耗。在 20 个客户场景下，所提出遗传算法的路由结果如图 11 所示。图中的每个蓝色点代表一个客户，旁边的数字表示需求 D_i ，中心的红色点代表仓库。

TABLE. 2 GENETIC ALGORITHM PARAMETERS

Parameters	Values
Map Width	1000m
Number of Customers	20
Number of Population	30
Weight Capacity	1.0kh
Maximum number of carried parcels for a UAV	3
Mutation Rate	0.6

表格 2. 遗传算法参数

参数	值
地图宽度	1000m
客户数量	20
人口数量	30
载重能力	1.0 千赫
无人机最大携带包裹数量	3
变异率	0.6

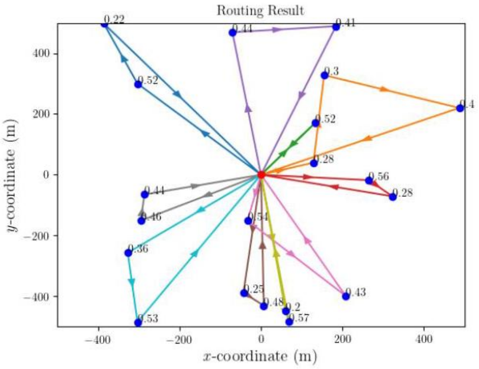

Fig. 11. Demonstration of the proposed genetic algorithm with 20 customers.

图 11. 在 20 个客户情况下所提出遗传算法的演示。

The convergence curve of a total of 100 customers with different population sizes is shown in Fig. 12. We can notice that the convergence rate is higher as the population size gets larger; however, the computation time also rises when the population size is increased. and green for 50 .

图 12 显示了总共 100 个客户在不同种群规模下的收敛曲线。我们可以注意到，随着种群规模的增大，收敛率提高；然而，当种群规模增加时，计算时间也会上升。绿色代表 50。

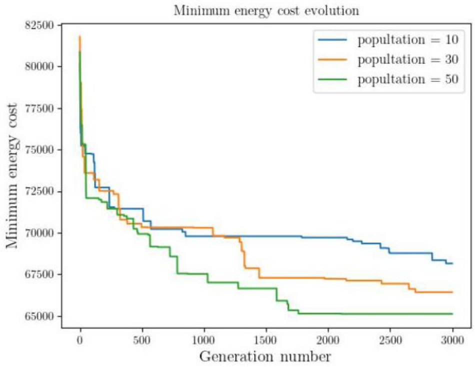

Fig. 12. Convergence curve of 100 customers concerning different population sizes. Blue stands for 10, orange for 30,

图 12. 100 个客户在不同种群规模下的收敛曲线。蓝色代表 10，橙色代表 30，

To evaluate the performance of the proposed genetic algorithm method, we compared the total energy cost produced by the genetic algorithm with the result using first come first serve scheduling (FCFS). Fig. 13 shows the improvement percentage when utilizing the proposed genetic algorithm with respect to FCFS in 100 missions. Each mission contains 20 customers. The experiment results show that the average improvement percentage is 29.41%.

为了评估所提出遗传算法方法的性能，我们比较了遗传算法产生的总能耗成本与使用先来先服务调度 (FCFS) 的结果。图 13 显示了在 100 次任务中使用所提出遗传算法相对于 FCFS 的改进百分比。每次任务包含 20 个客户。实验结果表明，平均改进百分比为 29.41%。

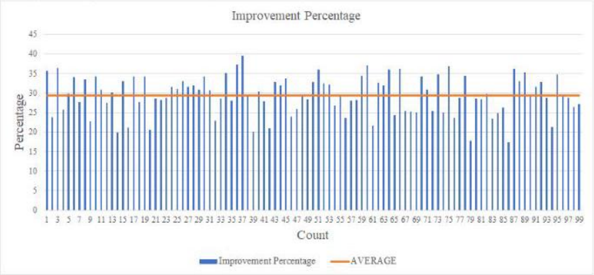

Fig. 13. Improvement our genetic algorithm in comparison to traditional FCFS method. We recorded a 29.41% improvement on average with respect to FCFS method.

图 13. 我们的遗传算法与传统 FCFS 方法的比较。我们记录了相对于 FCFS 方法的平均 29.41% 改进。

C. Adopting the Proposed Methods in a Simulated Parcel Delivery System

C. 在模拟包裹递送系统中采用所提出的方法

We now carry out simulation of the parcel delivery system. We first create a 3-D model of the campus of National Taiwan University (NTU) for the simulation, the model is shown in Fig.

我们现在进行包裹配送系统的模拟。首先，我们为模拟创建了一个国立台湾大学 (NTU) 校园的 3D 模型，模型如图所示。14.

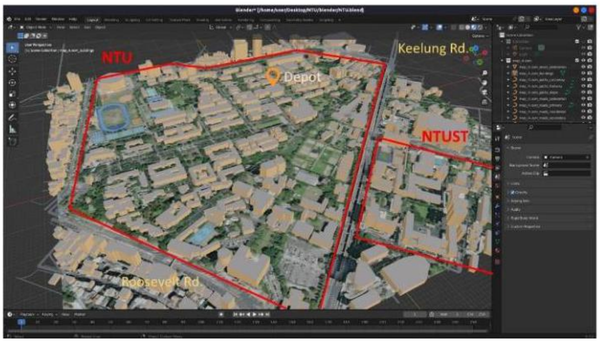

Fig. 14. 3-D model of the NTU campus in Blender

图 14. NTU 校园的 3D 模型在 Blender 中的展示

In the simulation, 20 orders with different parcel weight and destination are generated. Then, 10 UAVs will depart the central depot after being loaded, follow the route generated by our genetic algorithm to deliver the parcel to the right position while avoiding collision with each other, and finally go back to the original depot. Each UAV in the simulation can carry 1 kg of payload and 3 parcels at the maximum, flying at a height of $h = 100$ meters. The scene of UAVs taking off from the depot is shown in Fig. 15, and the route generated by our algorithm is shown in Fig. 16. In the end, it took 5 minutes and 8 seconds for 10 UAVs to deliver 20 parcels to customers at different locations. We now showed the feasibility of applying the proposed UAV collision avoidance method and the genetic algorithm based on route optimization in parcel delivery tasks with the result gained from the simulation.

在模拟中，生成了 20 个具有不同包裹重量和目的地的订单。然后，10 架无人机在装载后离开中心仓库，遵循我们的遗传算法生成的路线，将包裹送到正确的位置，同时避免相互碰撞，并最终返回原仓库。模拟中的每架无人机最多可携带 1 kg 的有效载荷和 3 个包裹，飞行高度为 $h = 100$ 米。无人机从仓库起飞的场景如图 15 所示，由我们算法生成的路线如图 16 所示。最后，10 架无人机用了 5 分 8 秒将 20 个包裹送到不同位置的顾客手中。我们现在展示了在包裹配送任务中应用所提出的无人机避障方法和基于路线优化的遗传算法的可行性，这是通过模拟得到的结果。

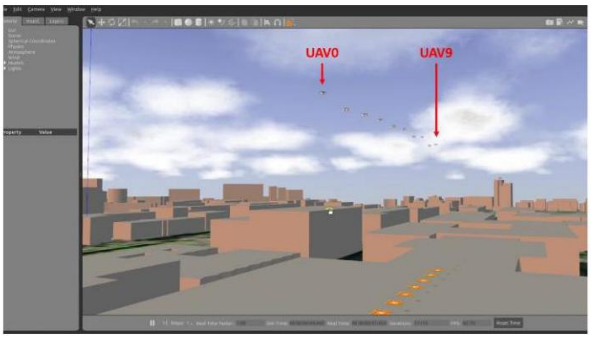

Fig. 15. Illustration of the UAVs taking off and start delivering.

图 15. 无人机起飞并开始配送的示意图

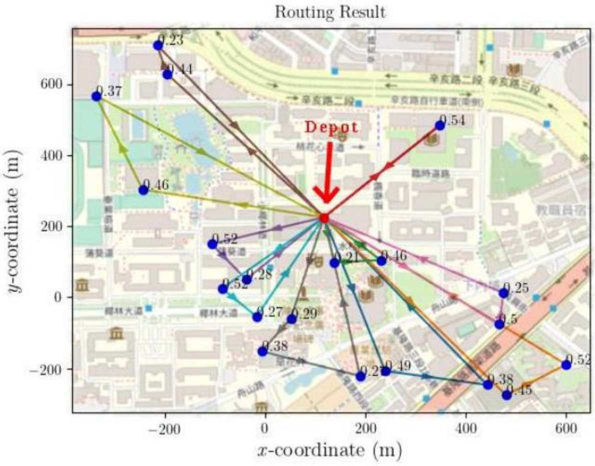

Fig. 16. Routing result of the simulation
图 16. 模拟的路线结果

VI. CONCLUSION AND FUTURE WORKS

VI. 结论与未来工作

In recent years, more logistics services are making use of the UAV parcel delivery system. However, implementing such a system involves solving some problems such as UAV collision avoidance and route optimization. Therefore, in this thesis, we proposed a UAV collision avoidance solution by adopting a state-of-the-art deep reinforcement learning model. Moreover, the proposed method works in a decentralized manner, it does not require successive communication with the centralized server, and it takes actions according to the local observation of an agent. In addition, to further reduce the cost of the logistic, we proposed a genetic algorithm to generate optimized delivery routes. Instead of minimizing the total distance, we aim to minimize the total energy cost in UAV delivery tasks. We modified the conventional CVRP with additional constraint and solve the optimization problem by using genetic algorithm along with our custom fitness function to obtain optimized routes in UAV delivery tasks. Finally, we validate the proposed method using physics engines and SITL to evaluate the feasibility of our proposed method. The result shows that our proposed UAV collision avoidance method is able to work in a realistic simulation environment. In future research, we aim to validate our UAV collision method on a UAV in a real-world environment. Also, we would like to compare the computational efficiency of the proposed genetic algorithm with some other modern non-linear programming solvers.

近年来，越来越多的物流服务开始利用无人机包裹递送系统。然而，实施这样的系统需要解决一些问题，例如无人机的避障和路线优化。因此，在本文中，我们提出了一种采用最新深度强化学习模型的无人机避障解决方案。此外，所提出的方法以去中心化的方式工作，不需要与中央服务器进行连续通信，而是根据代理的本地观察进行行动。另外，为了进一步降低物流成本，我们提出了一种遗传算法来生成优化的配送路线。我们不是最小化总距离，而是旨在最小化无人机配送任务的总能耗。我们对传统的 CVRP 进行了修改，添加了额外的约束，并使用遗传算法和我们的自定义适应度函数解决了优化问题，以获得无人机配送任务的优化路线。最后，我们使用物理引擎和 SITL 验证了所提出的方法，以评估我们提出方法的可行性。结果显示，我们提出的无人机避障方法能够在真实的模拟环境中工作。在未来的研究中，我们旨在在现实世界环境中的无人机上验证我们的无人机避障方法。此外，我们还希望将所提出的遗传算法的计算效率与其他现代非线性规划求解器进行比较。

REFERENCES

参考文献

[1] S. R. R. Singireddy and T. U. Daim, ”Technology roadmap: Drone delivery-amazon prime air,” in Infrastructure and technology manage- ment, Springer, 2018, pp. 387-412.

[2] I.Hong, M.Kuby, andA.T.Murray,”Arange-restrictedrechargingsta- tion coverage model for drone delivery service planning,” Transportation Research Part C: Emerging Technologies, vol. 90, pp. 198-212, 2018.

[3] S. M. Shavarani, M. G. Nejad, F. Rismanchian, and G. Izbirak, ”Ap- plication of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San

Francisco,” The International Journal of Advanced Manufacturing Technology, vol. 95, no. 9, pp. 3141-3153, 2018.

[4] H.HuangandA.V.Savkin,”Deploymentofchargingstationsfordrone delivery assisted by public transportation vehicles,” IEEE Transactions on Intelligent Transportation Systems, 2021.

[5] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, ”Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones,” IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10483-10495, 2019.

[6] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, ”Synchronized truck and drone routing in package delivery logistics,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 5772-5782, 2020.

[7] K.-W. Chen, M.-R. Xie, Y.-M. Chen, T.-T. Chu, and Y.-B. Lin, ”DroneTalk: An Internet-of-Things-Based Drone System for Last-Mile Drone Delivery,” IEEE Transactions on Intelligent Transportation Sys- tems, 2022.

[8] M. Quigley et al., ”ROS: an open-source Robot Operating System,” in ICRA workshop on open source software, 2009, vol. 3, no. 3.2: Kobe, Japan, p. 5.

[9] K.Loayza, P.Lucas, andE.Pelaez,”Acentralizedcontrolofmovements using a collision avoidance algorithm for a swarm of autonomous agents,” in 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), 2017: IEEE, pp. 1-6.

[10]D. Mellinger, A. Kushleyev, and V. Kumar, ”Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams,” in 2012 IEEE international conference on robotics and automation, 2012: IEEE, pp. 477-483.

[11] Y.-H. Hsu and R.-H. Gau, ”Reinforcement learning-based collision avoidance and optimal trajectory planning in UAV communication networks,” IEEE Transactions on Mobile Computing, vol. 21, no. 1, pp. 306- 320, 2020.

[12]D. Wang, T. Fan, T. Han, and J. Pan, ”A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing,” IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3098- 3105, 2020.

[13]X. Wang and M. C. Gursoy, ”Learning-based UAV trajectory opti- mization with collision avoidance and connectivity constraints,” IEEE Transactions on Wireless Communications, 2021.

[14]D.Silver, G.Lever, N.Heess, T.Degris, D.Wierstra, andM.Riedmiller, ”Determi nistic policy gradient algorithms,” in International conference on machine learning, 2014: PMLR, pp. 387-395.

[15]Y. Hou, L. Liu, Q. Wei, X. Xu, and C. Chen, ”A novel DDPG method with prioritized experience replay,” in 2017 IEEE international conference on systems, man, and cybernetics (SMC), 2017: IEEE, pp. 316-321.

[16]Y. Hou, J. Zhao, R. Zhang, X. Cheng and L. Yang, ”UAV Swarm Cooperative Target Search: A Multi-Agent Reinforcement Learning Approach,” in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 568-578, Jan. 2024.

[17] Y. Xue and W. Chen, ”Multi-Agent Deep Reinforcement Learning for UAVs Navigation in Unknown Complex Environment,” in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 2290-2303, Jan. 2024.

[18]L. Xu, T. Wang, J. Wang, J. Liu and C. Sun, ”Attention-Based Policy Distillation for UAV Simultaneous Target Tracking and Obstacle Avoidance,” in IEEE Transactions on Intelligent Vehicles

[19]Y. Hui, X. Zhang, H. Shen, H. Lu and B. Tian, ”DPPM: Decentralized Exploration Planning for Multi-UAV Systems Using Lightweight Information Structure,” in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 613-625, Jan. 2024

[20]C. C. Murray and A. G. Chu, ”The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery,” Transportation Research Part C: Emerging Technologies, vol. 54, pp. 86-109, 2015.

[21]Z. Wang and J.-B. Sheu, ”Vehicle routing problem with drones,” Transportation research part B: methodological, vol. 122, pp. 350-364, 2019.

[22]W. Yao et al., ”Evolutionary Utility Prediction Matrix-Based Mission Planning for Unmanned Aerial Vehicles in Complex Urban Environ- ments,” in IEEE Transactions on Intelligent Vehicles, vol. 8,

no. 2, pp. 1068-1080, Feb. 2023.

[23] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, ”Vehicle routing problems for drone delivery,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 1, pp. 70-85, 2016.

[24]N. A. Kyriakakis, T. Stamadianos, M. Marinaki, and Y. Marinakis, ”The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries,” Cleaner Logistics and Supply Chain, vol. 4, p. 100041, 2022.

[25]T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, ”Soft actor-critic: Off- policy maximum entropy deep reinforcement learning with a stochastic actor,” in International conference on machine learning, 2018: PMLR, pp. 1861- 1870.

[26]T. P. Lillicrap et al., ”Continuous control with deep reinforcement learning,” arXiv preprint arXiv:1509.02971, 2015.

[27]S. Fujimoto, H. Hoof, and D. Meger, ”Addressing function approxi- mation error in actor-critic methods,” in International conference on machine learning, 2018: PMLR, pp. 1587-1596.

[28]T. Haarnoja et al., ”Soft actor-critic algorithms and applications,” arXiv preprint arXiv:1812.05905, 2018.

[29]G. Brockman et al., ”OpenaAl Gym,” arXiv preprint arXiv:1606.01540, 2016.

Chun-Yuan Chi received M.S. degree in electrical engineering from National Taiwan University of Science and Technology in 2022. His research interests include unmanned aerial vehicles, vehicle path planning and machine learning.

程骏元于 2022 年获得国立台湾科技大学电气工程硕士学位。他的研究兴趣包括无人机、车辆路径规划和机器学习。

De-Fu Chen is current an undergraduate student of the mechanical engineering, National Taiwan University. His research interests include computer vision, intelligent vehicles and vehicle navigation.

陈德富目前是国立台湾大学机械工程的本科生。他的研究兴趣包括计算机视觉、智能车辆和车辆导航。

Hoang-Phuong Doan (Nikolas Doan) is a Research Assistant and Master’s Student in Control System Division, Department of Mechanical Engineering, National Taiwan University. His research interests include Smart Manufacturing, Additive Manufacturing, and Artificial Intelligence in Robotics.

黄方方 (Nikolas Doan) 是台湾大学机械工程学系控制系统分部的研究助理和硕士研究生。他的研究兴趣包括智能制造、增材制造以及机器人领域的人工智能。

Chung-Hsien Kuo received the Ph.D. degree in mechanical engineering from National Taiwan University, Taipei, Taiwan, in 1999. He is currently a Professor with the Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan. His main research areas include autonomous systems, novel soft robot design, sensors design, signal processing, computer vision and machine learning.

郭钟显于 1999 年在台湾台北的台湾大学机械工程系获得博士学位。他目前是台湾大学机械工程系的教授。他的主要研究领域包括自主系统、新颖的软体机器人设计、传感器设计、信号处理、计算机视觉和机器学习。