ORIGINAL RESEARCH **Open access**

Optimization of Truck–Drone Delivery Systems through Multi-Agent Reinforcement Learning

Thomas Dupont[1*], Marie Leclerc[1], Julien Martin[2], Claire Moreau[3]

Abstract

Lately, the use of cooperative truck-and-drone transportation has emerged as a novel tactic for boosting logistical productivity while reducing workforce depletion. This approach simultaneously addresses the battery-life limitations faced by drones and the delays trucks encounter during the final distribution segment. In this setup, trucks serve as mobile bases, warehousing platforms, and charging points for the drones. In light of the specific traits of collaborative truck–drone operations, this investigation constructed a simulation for multiple drone deliveries using the MPE library. Various optimization strategies were also implemented to boost algorithmic performance within this hybrid distribution system. The study concludes by benchmarking the proposed method against alternative multi-agent reinforcement learning techniques under identical conditions, verifying both the soundness of the problem's formulation and the excellence of the algorithm's results.

Keywords Reinforcement learning, Drone, Path planning, Multi-agent problem

*Correspondence:
Thomas Dupont
thomas.dupont@universite-paris-saclay.fr
[1] Department of AI for Grapevine Trunk Disease and Leaf Analysis, Faculty of Agriculture, Université Paris-Saclay, Paris, France
[2] Department of Precision Wind and Frost Protection Modeling, Faculty of Agriculture, University of Strasbourg, Strasbourg, France
[3] Department of Smart Soil pH and EC Monitoring, Faculty of Agriculture, University of Bordeaux, Bordeaux, France

Introduction

Collaborative truck–drone logistics refers to a system in which trucks serve as mobile bases and drones serve as the primary delivery mechanism. With the recent expansion of digital commerce and urban growth, alongside technological innovation and the rising cost of human labor, the global logistics sector has been seeking a more productive operational model [1]. Drones, or unmanned aerial vehicles, prized for their speed, efficiency, and adaptability, have attracted significant attention. As a result, the joint truck–drone delivery concept has been put forward and embraced by academic circles.

As depicted in **Figure 1**, while certain models and computational methods have probed the viability and time-saving potential of cooperative truck–drone deliveries—encompassing early trials and routing strategies by organizations such as FedEx, Meituan, and Amazon—the bulk of them have largely leaned on heuristic approaches for solitary vehicles or framed the drone as a parametric constraint. Systematic inquiry remains sparse, especially regarding coordinated truck–drone operations employing reinforcement learning techniques. As the logistical network architecture and drone navigation and command technologies advance, the practical design of cooperative truck–drones initiatives in targeted city sectors has become increasingly significant [2, 3].

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2023 The Author(s). | Published by Agro Intelligence Research Press



Figure 1. Truck–drone joint delivery.

Reinforcement learning has recently become a central topic of interest, applicable not only to systems control [4] but also to challenges in combinatorial optimization [5]. It works by allowing an agent to collect rewards from its interactions with the environment, then assess its action patterns. Through cycles of state-action evaluation and refinement, a Markov-based decision sequence is constructed to fulfill the supreme optimization goal. In the present research, we adopt reinforcement learning as the core framework for tackling a multi-vehicle cooperative truck–drone delivery scenario. Several algorithmic modifications have also been introduced. In addition, we test the resulting model against various other algorithms across multiple contexts, substantiating both the model's and the algorithm's effectiveness.

The present work addresses the path-optimization dilemma for multiple agents in a purely cooperative game framework by dividing the truck routing phases and drone flight corridors. Multi-agent algorithmic approaches were deployed to meet this aim. Factors such as the number of drones and clients serve as modifiable environmental inputs. This method successfully demonstrates that, within a multi-agent framework, state partitioning and fine-tuning of algorithmic hyperparameters can be broadly applied to resolve truck–drone routing optimization tasks.

Merging reinforcement learning with a tailored environment is key to the agile tuning of variables such as fleet size and delivery destinations [6]. By establishing varied settings with distinct vehicle and client totals, we can emulate and confirm the model's effectiveness across different population density configurations. Especially when dealing with multiple drones, the game-theory-informed fully cooperative stance enables us to reframe the multi-agent challenge as a single-agent decision task. This tactic shortens lengthy training periods and lessens the burden of navigating an extensive combinatorial action space.

Additionally, the randomized exploration mechanism used in this investigation finds optimal solutions with greater ease than classic heuristic methods, sidestepping the need for additional parameter calibration and encoding-decoding workloads.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

This manuscript introduces a learning environment derived from an authentic delivery scenario that integrates multiple goals and time constraints, built on a foundation of mixed multi-truck and multi-drone dispatch situations. We tackle this issue through the lens of reinforcement learning, centering on the analysis of drone flight paths within a truck–drone delivery paradigm. The model covers features such as vehicle departure points, fleet size, client count, positional data, and time. To address the challenges of an enormous shared action domain and the potential for reward competition or friction introduced by reinforcement learning in multi-agent settings, we apply a multi-objective framework. Expanding on single-agent methods, we segment the vehicle routing problem into discrete stages within a single simulation run, efficiently addressing shortcomings typical of standard truck-based distribution. The central aim is to curtail total drone airtime while maximizing the reward for serving clients, all under defined time boundaries. Through reinforcement-learning-enabled route refinement under varied parametric conditions inside a consistent environment, we eventually identify a suite of promising near-optimal solutions.

The real-world value of this research lies in its extension and refinement of delivery frameworks and single-agent computational solutions for multi-agent teamwork. High workforce costs, poor throughput, and limited agility for swift urban deployment beset established distribution models. Viewing the truck–drone dispatch problem via multi-agent algorithms uncovers several fundamental hurdles. To begin with, scaling the number of agents—both vehicles and customers—is unwieldy and inflexible. Following that is the challenge of formulating joint action mappings; in a multi-agent context, each agent's moves affect the others, leading to fluctuating reward signals and unstable state dynamics. Finally, embracing truck–drone delivery configurations can markedly improve distribution efficiency in city environments, slashing temporal waste and personnel costs. This has particular value in reducing the physical burden on the logistics workforce. Aligned with the hierarchical depot distribution networks currently favored in numerous metropolitan areas, this model reduces intermediate handling steps and disruptions throughout the delivery chain.

Furthermore, truck–drone transport can assist operations requiring high-speed delivery, such as dispatching rapid, accurate shipments to maritime cargo firms at ports via combined truck–drone channels [7, 8]. To conclude, the truck–drone delivery paradigm marks a key evolutionary path for the logistics industry's future and carries the potential to become an emblematic mode of next-generation logistics. It is positioned to support economic expansion and social betterment.

This article delivers the following targeted contributions: (1) It examines the hybrid truck–drone distribution challenge through the application of reinforcement learning, thereby extending scholarly work on truck–drone operational models. (2) It refines both algorithms and simulation settings for multi-agent conditions, applying Monte Carlo Tree Search as an enhancement mechanism. The methodology splits the vehicle routing challenge into phases to accelerate the establishment of truck routes and applies multi-agent reinforcement learning methods to drone flight routing. Regularization procedures are harnessed to strengthen learning speed and consistency. (3) It builds an adaptable environment anchored in the truck–drone delivery concept, permitting the manipulation of factors like the number of drones and clients. This testbed supplies a rich source of experimental setups for subsequent truck–drone delivery investigations.

Related Work

Goods movement and distribution have long stood as a research priority for both scholarly circles and the commercial world. Over recent years, the booming e-commerce and meal-delivery sectors have driven numerous firms and academic bodies to design and roll out a range of novel distribution schemes. Currently, the majority of city logistics relies on multi-echelon depot systems, where parcels are routed through successive levels of hubs, starting from upstream centers and moving down to subordinate nodes until they hit the "last-mile" hurdle. This arrangement results in a heavy drain on human capital and generates many unnecessary intermediary links. As presented in **Table 1**, despite certain studies and pilot projects targeting the "last-mile" issue and logistical reconfiguration, identifying a root-level fix within conventional delivery frameworks remains elusive.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Table 1. Research direction.

Study	[9]	[10]	[4]	[11]	This Study
Algorithm	Heuristic method	MAPPO	MARL	AC & DQN	MAPPO
System model	MTSP	Nash equilibrium framework	Mean-field model	FSTSP	Hierarchical model
Optimization objective	Profit maximization	Communication efficiency	Multi-UAV coordination	Delivery time minimization	Profit & time optimization

With the maturation of drone technologies and adjacent domains such as power storage, UAVs have experienced extensive adoption over the past 10 years across a multitude of industries [12], including maritime freight [13, 14], protective services, package conveyance, volumetric reconstruction, crisis response, and defense functions. The issue of drone-based dispatching can be perceived as a derivative of conventional routing challenges and aerial guidance disciplines. When measured against standard delivery vehicles, drones offer greater agility, airborne operability, and faster launch times, positioning them as a highly enticing tool in the distribution realm. Despite enduring endurance restrictions and legislative hurdles, these impediments have been partly mitigated through state-imposed rules and the use of trucks as mobile launch bases [15]. The hybrid truck–drone model draws upon the combined virtues of both transport modes. Trucks efficiently counteract drones' deficiencies in operational reach and freight capacity, whilst drones supplement trucks in the final delivery stretch, especially in dense metropolitan zones characterized by skyscrapers, thereby eliminating delays and manpower waste. This renders truck–drone logistics an appealing paradigm distinguished by swiftness, affordability, and configurability [16].

Collaborative truck–drone logistics further entails a separate set of roles and stipulations relative to orthodox vehicular delivery systems. Most critically, in terms of routing, both terrestrial transport and aerial craft must factor in the transit pathway. In tandem, charting drone flight corridors involves accounting for a host of parameters, above all blockages and spatial gaps when several drones are airborne concurrently; an even richer set of factors comes into play, necessitating elaborate decision-making to handle the reciprocal effects among the trajectories of varied drone formations. Beyond these facets, the adoption of reinforcement learning methodologies enables a more detailed analysis of the truck–drone routing conundrum. It empowers scholars to run multifaceted simulations, including scenarios where actual experimental conditions cannot be replicated. Single-agent paradigms nonetheless bump up against constraints when navigating the action and state topographies peculiar to multi-agent milieus. Consequently, a more versatile and universally applicable scheme is called for to meet the requirements of multi-drone flight-path formulation and truck itinerary structuring in joint truck–drone undertakings. To sum up, as a trailblazing logistical methodology, hybrid truck–drones need deeper trialing and polishing before they attain market-readiness and on-the-ground deployment.

Casting classic heuristic strategies as a benchmark, the recent period has seen substantial scrutiny of cooperative drone employment for terminal-leg distribution, as typified by the contributions of Freitas *et al.* [17] on the truck–drone delivery problem. Their publication unveiled a new Mixed-Integer Programming (MIP) formulation, along with a heuristic approach to address the problem. In every tested case, the furnished MIP framework yielded higher linear relaxation thresholds than prior formulations and optimally solved multiple previously unsolved benchmark instances from the academic literature. The undertaking applied a blended heuristic procedure anchored in a generalized variable neighborhood search, fused with aspects of a taboo search, to produce high-quality results for large test cases. Widespread experimentation across multiple instances attests to the potency of this scheme.

In a distinct investigation, the authors used machine learning to optimize truck–drone dispatch by Chang and Lee [18], focusing on pinpointing an efficient transit loop for a drone-bearing truck. After preliminary K-means clustering and Traveling Salesman Problem (TSP) abstraction, the inquiry aimed to determine offset coefficients for repositioning cluster

© 2023 The Author(s). | Published by Agro Intelligence Research Press

nuclei, thereby extending the drone service radius along compact truck pathways. To substantiate the strength of the suggested formulation incorporating transfer weights, it was measured against two separate delivery routing methodologies. Empirical findings from paired t-tests on randomly generated delivery coordinates showed that the proposed scheme outperformed the other two formulations.

Kitjacharoenchai *et al.* [19] and Moshref-Javadi *et al.* [20] merged canonical puzzles and undertook the TSP, incorporating a lone drone alongside a truck. Kitjacharoenchai *et al.* [19] posited that the truck could pause at patron positions and dispatch the drone repeatedly while the truck remained immobilized to deliver goods. This amplified the Traveling Repairman Problem (TRP), and a fused tabu search simulated annealing routine was harnessed to crack the problem.

Factoring in governance and climatic variables, Jeong *et al.* [21] integrated the effect of package heft on drone battery drain and off-limits airspace. Specifically, the flight envelope of drones proved acutely sensitive to payload mass, and drones faced restrictions on overflying classified sites overseen by the Federal Aviation Administration (FAA) within the United States, as well as transitory prohibitions in particular zones prompted by meteorological conditions. The inquiry erected a mathematical edifice incorporating those elements and rolled out a two-stage construction procedure coupled with a search heuristic to enhance computational tractability for real-world problems.

Regarding reinforcement learning, Liu *et al.* [11] broadened the Flexible Service Technician Scheduling Problem (FSTSP) to encompass probabilistic journey spans and redefined it as a Markov Decision Process (MDP). This representation was resolved using reinforcement learning methods, notably Deep Q-Networks (DQN) and Advantage Actor-Critic (A2C) algorithms, to avoid the combinatorial growth of the state space. Corroboration was undertaken using artificially generated datasets, and the results established that, even in the presence of stochastic journey durations, reinforcement learning techniques eclipsed the efficacy of approximate optimization schemes and outstripped MIP constructs and localized search heuristics on the foundational FSTSP.

The preceding publications largely rely on established heuristics, such as genetic algorithms, to address optimization challenges in joint truck–drone distribution. In contrast, the present manuscript leverages reinforcement learning with the MPE library to refine the truck–drone delivery formulation under a multi-objective schema parameterized by client and vehicular attributes. When juxtaposed with classical methodologies, the solution advanced herein places greater weight on tangible imperatives, such as fleet timing and schedule harmonization. Moreover, by infusing the multi-agent problem, adaptations can be calibrated across dissimilar vehicular counts, steering the resultant findings closer to real-life operational contexts.

Currently, routing problems in purely truck-based frameworks, where duration serves as the cardinal benchmark, are labeled MLP or TSP. This matter is commonly referred to as the Customer-Centric Routing Problem and typically takes the form of a single-criterion problem that minimizes time. Keeping in mind the idiosyncratic advantages of drones for enabling swifter deliveries, this paper highlights the multi-agent problem, a challenge addressed through incentive architectures and restrictive constraints to more faithfully reflect the complex nature of real-world settings [22].

The following sections of this paper are organized as follows: Section 2 sketches the mathematical formulation and the problem's virtual testing bed. Section 3 acquaints the reader with the reinforcement learning frameworks and cardinal conceptual definitions. Section 4 stages a comparative evaluation of the framework's competence. Section 5 rounds off the paper and charts prospective trajectories for future inquiry.

Problem Description

This segment outlines the divergences between the delivery architecture proposed in this study and standard truck-oriented logistics frameworks. A granular breakdown of the model's underlying assumptions is provided, along with a delineation of how multi-agent and single-agent reinforcement learning diverge when deployed in the truck–drone operational context.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Truck–drone logistics embodies a stratified hybrid dispatching blueprint, typically comprising several trucks, each carrying a set of drones, as depicted in **Figure 2**. Within this blueprint, drones serve as delivery instruments, replacing human couriers, while trucks handle a constellation of duties—inventory holding, operational coordination, maintenance, launch, and power replenishment. Nascent drone delivery paradigms hinged principally on ousting human workers from the terminal delivery phase to slash payroll outlays. Still, mismatches in delivery time windows led to significant drone idle time. On top of that, the hard ceiling on drone flight distance restricted them to a disc-shaped operating zone whose midpoint was at the dispatch depot and whose perimeter was defined by the maximum flight radius. This unintentionally inflated the investment needed to roll out a blanket delivery grid across cityscapes.

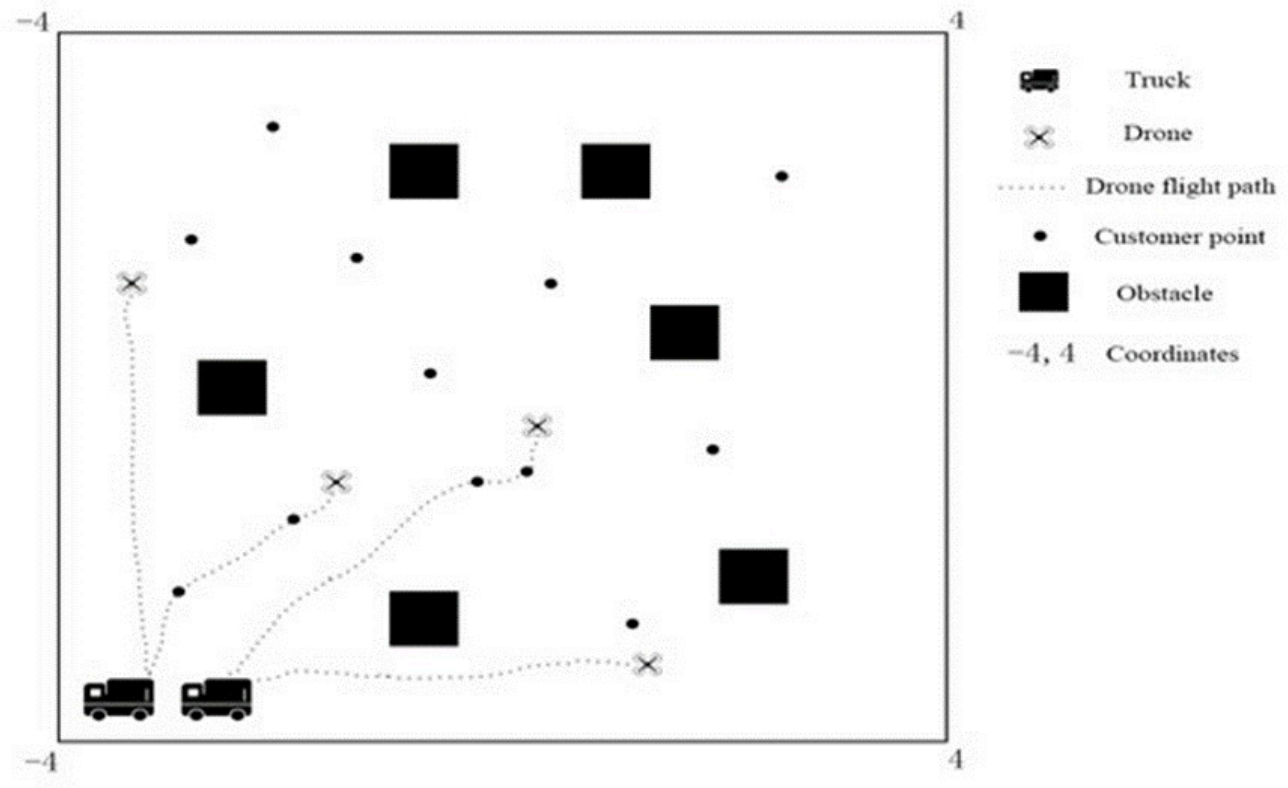

Figure 2. Simulated environment for an 8 × 8 square mile area.

Throughout this investigation, drones are framed as orbiting the trucks, carving out a disc-shaped service footprint with the peak flight span serving as the radius. Crucially, this disc-shaped footprint is free to shift along with the trucks, and the trucks themselves serve as versatile staging platforms that aid the drones in fulfillment tasks. This configuration measurably lifts delivery throughput while compressing the capital required for logistics network construction.

Across this study, the coordination of a fleet of trucks and drones is organized within a graph-structured domain, with client service nodes serving as graph vertices. This formulation mirrors actual metropolitan residential geographies and proves advantageous for streamlining both the mathematical abstraction and the state portrayal demanded by reinforcement learning.

The present work concurrently engages multiple trucks in fulfillment operations, and the objective landscape comprises several simultaneous goals. Furthermore, the decision-making entities in this model engage in purely symbiotic interactions, closely resembling cooperative game-theoretic arrangements. To heighten the yield of the reinforcement learning procedure, the research incentivizes decentralized action selection through a unified overarching target. The schooling of agents proceeds in a sequential cascade, in which the action-history trace generated by a preceding agent serves as the learning substrate for the successor agent. This staging mechanism compresses the learning ramp-up

© 2023 The Author(s). | Published by Agro Intelligence Research Press

interval of the reinforcement learning algorithm. As for drones, the study abstracts away granularities of the flight dynamics and distills service nodes using off-the-shelf drone trajectory-planning software alongside satellite imagery. This distillation reduces the path-parameter space, especially when factors such as wind patterns and built-environment obstacles are considered.

To crack the truck–drone fulfillment puzzle from a multi-agent perspective, this work leverages variants of multi-agent deep reinforcement learning, such as Multi-Agent Proximal Policy Optimization (MAPPO) and Multi-Agent Deep Deterministic Policy Gradient (MADDPG). By deploying these multi-agent schemes and adopting a consolidated decision-processing posture in a fully cooperative mode, the study effectively replicates and untangles the truck–drone logistics challenge through the prisms of game-theoretic reasoning and mathematical formalization.

Notations and model

This subsection rolls out the optimization formulation and the symbolic shorthand embedded within it. The formulation draws on fragments of selected algorithmic investigations related to the TSP [23, 24]. That said, building on the algorithmic machinery and simulated backdrops advanced in this manuscript, we have broadened and reworked select equations and definitions to better align with the fulfillment challenge arising in the truck–drone cooperative setting.

Mathematical notations

1. P The set of parking points P = {1, 2, . . . , p}.
2. C The set of customers C = {1, 2, . . . , c}.
3. U The set of drones U = {1, 2, . . . , u}.
4. V The set of trucks V = {1, 2, . . . , v}.
5. t Unit time. It can be regarded as a step of the algorithm in this study.
6. v^t Truck speed.
7. v^d Drone flight speed.
8. Cp The set of internal customer numbers at parking points p, where $p \in$ P.
9. $d^V_{i,j}$ The distance of the truck from parking point i to parking point j, where $i, j \in$ P.
10. $d^U_{i,k}$ The distance of the truck from parking point i to customer k, where $i \in$ P, $k \in$ C.
11. $r_{i,p}$ The reward received from customer i at parking points p, where $i \in$ C, $p \in$ P.
12. $t_i^T{}_{,j} = d^V_{i,j}/v^t$; the travel time of the truck from point i to point j, where $i, j \in$ P.
13. $t_i^D{}_{,k} = 2 * d^U_{i,k}/v^d$; the flight time of the drone from parking point i to customer k, where $i \in$ P, $k \in$ Cp.
14. t^w_p Total time spent at parking point p, where $p \in$ P.
15. P_i If parking point i is visited, $P_i = 1$; otherwise, $P_i = 0$. $i \in$ P.
16. C_k Total time accumulated before serving customer k, where $k \in$ C.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

17. R_p Total customer reward at parking point p, where $p \in$ P.

Decision variables

1. $V_i^n{}_{,j}$ If truck n moves from node i to node j, then $V_i^n{}_{,j}$ is equal to 1, otherwise it is 0,

$\forall n \in$ V

2. $U_i^n{}_{,j}$ If drones n moves from customer i to customer j, then $U_i^n{}_{,j}$ is equal to 1, otherwise it is 0, $\forall n \in$ U.

Objective functions

$$min\ C_k \quad C_k = \sum_{t\in\mathbb{P}}\sum_{f\in\mathbb{P}}\left(t_{t,f}^{T} * V_{t,f}^{n}\right) + \sum_{t\in\mathbb{P}} P_t * \left(\sum_{k\in C_p}\sum_{t=1}^{k-1} t_{t,k}^{D} * U_{t,f}^{n}\right)$$

$$max\ R_p \quad R_p = \sum_{t\in\mathbb{P}} P_t * \sum_{t\in C_p} r_{t,p}$$

(1)

Constraints

(1) To prevent later-stage scheduling snarls, the travel-time spread between any two trucks shall not exceed 30%.

$$0.7 \le \sum\nolimits_{v\in\mathbb{V}} \left(t_{t,f}^{T_{v1}} / t_{t,f}^{T_{v2}}\right) \le 1.3 \qquad (2)$$

(2) The order payoff will decay to some extent as the clock advances, yet it is bounded below by zero.

$$r_{t+1,p} = r_{t,p} - f\left(C_k\right) \qquad \forall r_{t,p} \ge 0 \qquad f\left(C_k\right) = C_k - C_k\left(t_p^w / 8T\right)$$

(3)

(3) Any single drone can be tied to only one truck at any given instant.

$$\sum\nolimits_{n\in\mathbb{U}}\sum\nolimits_{t\in\mathbb{C}}\sum\nolimits_{j\in\mathbb{C}} U_{t,f}^{n} \le 1 \quad \forall i \in \mathbb{C} \quad \forall j \in \mathbb{C} \qquad (4)$$

(4) Every halt location shall be touched by a truck at least once.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\sum_{j \in \mathbb{P}} V_{i,j}^{n} = 1 \quad \forall n \in \mathbb{V} \quad \forall i \in \mathbb{P} \quad (5)$$

(5) Preserve alignment between truck and drone displacements. Whenever a truck is underway from halt i to halt j, the drone is to shift from client i to client j.

$$V_{i,j}^{n} = U_{i',j'}^{n} \quad \forall n \in \mathbb{U} \quad \forall i,j,p \in \mathbb{P} \quad \forall i',j' \in \mathbb{C}_{p} \quad (6)$$

Objective function (1) is sculpted around the aim of shrinking delivery duration, predicated on trimming client queuing time. Objective function (2) attaches to the underlying worth of client orders or prospective payoffs, which can be interpreted as gratuities or contentment scores, and it wanes according to a function tied to waiting duration, without ever slipping below zero. In tandem, these two objective functions serve as the yardsticks for this study's optimization. In this work, set in a real-world setting, it is mandated that a drone attend to each client and that each truck stop at every delivery node to guarantee universal service coverage. Constraint (1) is intended to handle scheduling hiccups that could arise from stark asymmetries in operating hours across the truck fleet, even when several trucks are fielded. Constraint (2) codifies that every delivery node is reached, thereby upholding service for every client. Constraint (3) pins down client payoffs, modulated by order valuation and delivery timing.

Reinforcement learning for truck–drone delivery

This subsection provides an overview of the algorithmic components and their operationalization in our study to address the truck–drone distribution problem. We foreground the harnessing of reinforcement learning paradigms for itinerary refinement inside the cooperative truck–drone framework. In addition, we cast a spotlight on the embedding of multi-agent algorithms to steer drone flight corridors and unpack the weaving in of Monte Carlo Tree Search as a probe mechanism to polish truck itineraries. As a supplementary step, we apply reward-environment normalization procedures to improve the algorithm's speed and convergence to optimal solutions. Beyond this, we conduct comparative evaluations across diverse algorithms and architectural blueprints to substantiate the effectiveness of our formulation and reinforcement learning pipelines.

Reinforcement learning, drawing on behaviorist psychological theories, serves as a cornerstone of learning, sharpening an agent's behavioral policy through feedback loops with its environment to maximize cumulative rewards. In multi-agent regimes, reinforcement learning can be divided into collaborative and adversarial modes, each aimed at maximizing either shared payoffs or rivalrous outcomes across a population of agents. Scholarship on multi-agent reinforcement learning fans out into game-theoretic abstractions that accommodate the tangled interdependencies among intelligent actors and taps into deep reinforcement learning machinery, wielding neural architectures to amplify decision-making prowess. These investigatory streams aim to equip reinforcement learning collectives to mesh more fluidly within complex environments and to hone their decision-making sequences to attain higher reward totals or carry out more intricate missions.

In our inquiry, we added a supplementary layer of randomized exploration guided by Monte Carlo Tree Search during the opening rounds of the reinforcement learning cycle, especially within the vehicle-routing layer of the environment. This adjunct works to fast-track the familiarization of intelligent agents with their operational milieu while shielding them from settling into suboptimal basins of the solution landscape.

The interleaving of neural network approximators with reinforcement learning, above all in deep reinforcement learning, opens an avenue to tame the "curse of dimensionality". Throughout this investigation, deep reinforcement learning is

© 2023 The Author(s). | Published by Agro Intelligence Research Press

identified as the primary focus of our experimental agenda. Going a step further, the present research draws on third-party libraries, namely Gym and the MPE toolkit (built on Gym), to enable richer algorithmic benchmarking. The experimental sandbox is purpose-built with tunable knobs, opening the door to cross-scenario testing that highlights the robustness of both the modeling framework and the algorithmic machinery.

Within the environmental specification outlined in this study, a collection of intelligent agents operates through a phased decision-making process. Once the lead intelligent agent has probed and internalized the environment, weight assessments based on action trajectories are performed. This sequence embeds the behavioral footprint of the lead intelligent agent into the updated environment's constitution. At the endpoint, in a fully cooperative game configuration, centralized information handling is applied to multi-agent challenges. This tactic meaningfully pares back the overhead of constructing joint-action mappings and sharply condenses the multi-agent state representation. It also curtails exploration and assimilation runtimes, thereby greatly elevating the viability of reinforcement learning for orchestrating truck–drone delivery plans within city boundaries.

On the side of mathematical abstraction, this investigation uses the Vehicle Routing Problem (VRP) as its scaffolding, reconfiguring it to accommodate vehicles and service nodes while simultaneously introducing drones as novel decision variables. The study aims to align the mathematical formalization more closely with real-world multi-vehicle logistics scenarios and to broaden its generality to meet the demands of a range of actual application domains. Even as it zeroes in on disentangling the VRP, the injection of drones as an inventive parameter widens the aperture of use cases accessible to the research community. The introduction of this model carries substantial scholarly and pragmatic import for tackling knotty supply chain, logistics, and transportation puzzles encountered in practice.

This paper first delineates a multi-objective problem scoped by time-based bounds tied to the objective function. From a logistics perspective, both clock time and operational throughput rank among the most consequential optimization targets. Furthermore, serving as a stand-in for multi-echelon distribution architectures, we must also incorporate client satisfaction into final-leg deliveries. Therefore, we position vehicle travel duration and client payoffs as the primary optimization objectives, aiming to boost systemic efficiency at the macro level and service quality at the micro level. Finally, in our capacity as delivery service operators, we shoulder the obligation to attend to every client, compelling vehicles to call at every stopover node—this forms the very substrate of the TSP. In the matter of inter-vehicle orchestration, we factor in the imperative of balanced wrap-up times among trucks while servicing city districts, which gives rise to constraints that enforce parity in vehicle completion times. From the vantage point of drones, we use multi-agent reinforcement learning to coordinate drone teamwork during delivery and incorporate environmental factors such as mid-air collisions and physical obstructions.

Principle and Application

Policy-based reinforcement learning and policy gradient

The principal algorithmic workhorse underpinning the present study is MAPPO, a deep reinforcement learning offshoot rooted in the AC architectural paradigm. The intellectual lineage of Actor–Critic stretches back to the seminal work of Barto *et al.* [25] in 1983, which laid the foundation for the modern AC blueprint now widely in circulation. AC resides within the family of policy-conditioned reinforcement learning strategies. Within this bifurcated design, the actor module performs direct environmental interaction, whereas the critic module serves as a scoring referee, refreshing the associated value or policy approximators. The parameter refresh cycle unfolds as follows: the actor roams the state space, yields experiential samples, and derives gradient signals. Those gradient signals are then passed to the critic, which recalibrates its internal weights using methods such as stochastic gradient descent and backpropagation. Once the weight recalibration completes, the revised parameters flow back to the coupled actor in preparation for the subsequent round of environment sampling. Across the lion's share of evaluation settings, AC-flavored algorithms exhibit several attractive properties, not least improved convergence properties and superior efficacy in high-dimensional, continuously valued action spaces.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

These properties map cleanly onto the aims and constraints framing this work, easing the path through the optimization solve pipeline.

The neural network substructure woven into the AC formulation further amplifies storage and representational throughput, helping tame the “curse of dimensionality” and broaden the algorithm’s portability. In the discussion that follows, we briefly dissect the inner mechanics of the AC algorithm and probe its prospective role and comparative strengths within this inquiry. In addition, the AC family tree has branched into numerous variants and antecedent forms, among them the A3C formulation and its multi-agent sibling MA-A3C. A sizable collection of other multi-agent constructs similarly borrows architectural motifs from the AC template. Adopting this algorithmic lineage as a testbed creates a favorable platform for subsequent benchmarking against competing multi-agent formulations.

The AC (Actor–Critic) scaffold is policy-grounded, and its earliest recognizable form was the QAC algorithm, itself a lineal descendant of the policy gradient (PG) lineage. This scaffold constitutes the archetypal mold from which the overwhelming majority of policy-conditioned reinforcement learning algorithms are cast. The Actor–Critic construct can be cleaved into two functional halves. The Actor half, issuing from policy gradient ancestry, is endowed with the capability to draw suitable actions inside continuous action manifolds—a competence notably absent from value-grounded Q-learning, which remains largely tethered to discrete action sets. On the other side of the coin, the Actor’s learning pace is sluggish because its updates hinge on the aggregate return across an entire episode. At this crossroads, splicing in a value-grounded module as the Critic unlocks the door to single-step updating via Temporal Difference machinery, in effect trading away some degree of bias to buy down variance. This reciprocal, mutually beneficial interplay between the two halves provides the foundation for the Actor–Critic architecture. Under this architecture, the Actor pulls actions from a probabilistic distribution, the Critic provides a quality score for the Actor’s actions, and the Actor subsequently tweaks its action-sampling likelihoods in response to the Critic’s feedback.

In keeping with other reinforcement learning formalisms, AC algorithms operate with elementary building blocks—time (t), states (s), rewards (r), and actions (a)—to characterize their operational environment. Within policy-grounded reinforcement learning and AC algorithms, the tuning of policy parameters typically rests upon the gradient of some scalar policy performance measure (θ). Because these techniques aim to maximize this measure, their parameter adjustments approximate a climb along the gradient of J.

$$\theta_{t+1} = \theta_t + \alpha \nabla \widehat{J(\theta_{t+1})} \quad (7)$$

Within this expression, $\nabla \mathrm{J}\,(\widehat{\theta \mathrm{t}+1})$ serves as an approximate estimation of the gradient associated with the performance measure parameter θ.

The integration of a state–value function arms AC algorithms with the capacity to mount a supplementary appraisal of action quality. In the TD(0) variant of the AC formulation, a differentiable parameterized policy $\pi(a|s,\theta)$ is accepted as input alongside a differentiable parameterized value function $v(s,w)$. The learning regime entails the cyclical refinement of both the initial policy parameters θ and the weights w tied to the state–value function. For each non-terminal state S encountered within every subsequence, an action A is sampled. The ensuing jump to the successor state S' after executing action A is registered, whereupon both the weights w and the policy parameters θ undergo a revision step. This iterative loop enables the assessment of the value of state S.

In continuing problem formulations that lack subsequence delimiters, the policy is instead characterized through the mean return accumulated per time step t.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\begin{aligned} J(\theta) \quad &\doteq r(\pi) \doteq \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}\left[R_t | S_0, A_{0:t-1} \sim \pi\right] \\ &= \lim_{t\to\infty} \mathbb{E}\left[R_t | S_0, A_{0:t-1} \sim \pi\right] \\ &= \sum_{s} \mu(s) \sum_{a} \pi(a|s) \sum_{a} p(s', r|s, a)\, r \end{aligned} \tag{8}$$

Where μ captures the stationary visitation distribution emerging under policy π, $\begin{matrix}\mu(\mathrm{s}) \\ \doteq \lim_{\mathrm{t}\to\infty} \Pr\{\mathrm{St} \\ = \mathrm{s}|\mathrm{A}_{0:t-1} \\ \sim \pi\}\end{matrix}$, which is taken to exist and to be independent of the starting state $S0$. In a manner paralleling problems delimited by subsequence boundaries, within continuing formulations, we call upon the differential return to specify the value function.

$$\sum_{s} \mu(s) \sum_{a} \pi(a|s, \theta)\, p(s', r|s, a) = \theta(s') \qquad \forall s' \in S \tag{9}$$

Concurrently, even though policy-conditioned reinforcement learning can sometimes trail value-conditioned rivals in end-stage performance when a solitary objective function governs—on account of the lethargic convergence behavior of policy updates—policy parameterization holds a pivotal theoretical advantage over action–value-grounded alternatives. Policy parameterization provides an explicit analytical expression that lays bare how the performance yardstick is modulated by policy parameters, completely bypassing any dependence on derivatives of state visitation distributions. This expression provides the theoretical underpinning for all policy gradient schemes.

As portrayed in **Figure 3**. Inside the PG algorithm, the Agent is also termed the Actor. Each Actor maintains its own policy π customized for a given task, typically instantiated as a neural network governed by parameters θ. Commencing from a particular state and persisting until the task wraps up—a span referred to as an episode—at each increment, a reward r is harvested; the summed reward realized for completing the task is denoted R. In this manner, an episode stretching across T time increments sees the agent interweaving with the environment, giving shape to a trajectory sequence τ, as laid out below:

$$\tau = (s_1, a_1, s_2, a_2, \ldots, s_T, a_T) \tag{10}$$

© 2023 The Author(s). | Published by Agro Intelligence Research Press

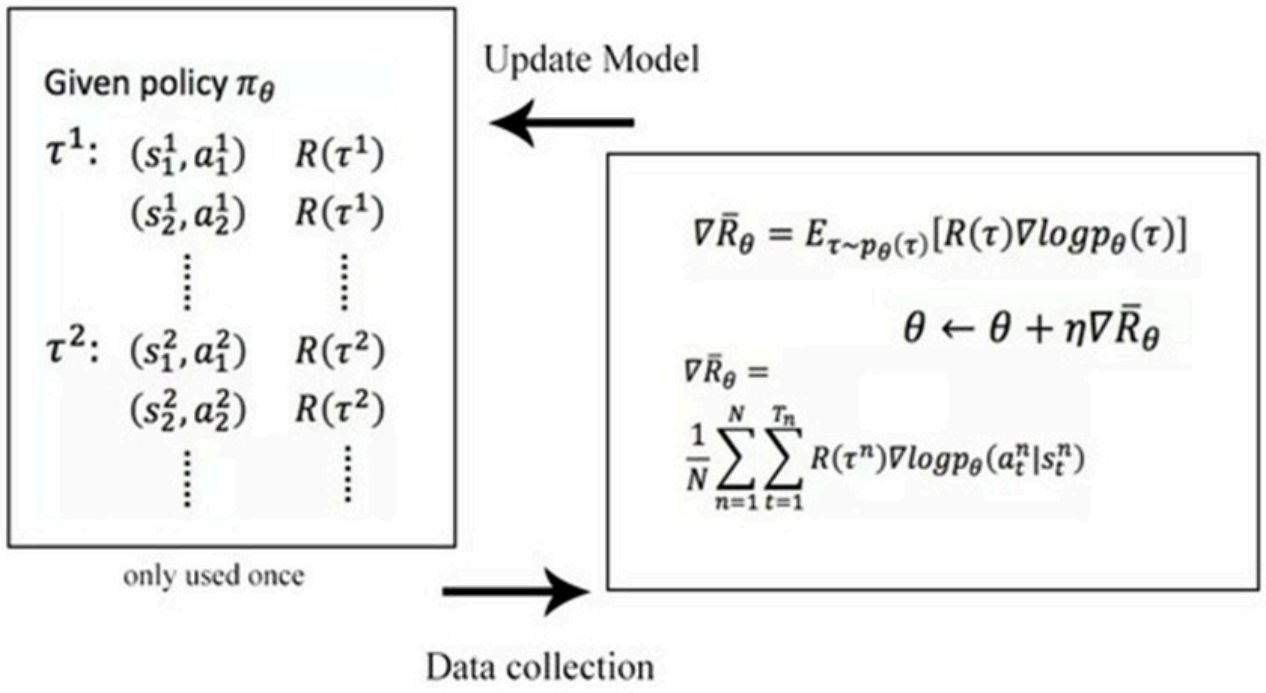

Figure 3. Policy gradient.

The aggregate reward collected over the sequence τ amounts to the summation of rewards reaped at each increment, represented as (τ). Accordingly, under the Actor's policy π, the expected reward attainable is:

$$\overline{R}_\theta = \sum_\tau R(\tau) P_\theta(\tau) = E_{\tau \sim P_\theta(\tau)}[R(\tau)] \quad (11)$$

The mandate placed upon the algorithm is to sculpt the Actor's policy π so as to push the expected reward to its ceiling. This is realized through the policy gradient machinery, which harnesses gradient ascent to nudge the network parameters θ (and thus reshape the policy π). In the final analysis, the algorithmic emphasis lies in unearthing parameter gradients.

Beneath the AC canopy, the Trust Region Policy Optimization (TRPO) algorithm was born, erected atop a natural policy gradient optimization scheme, and it rapidly drew widespread interest. Although natural policy gradient optimization helps polish policies in reinforcement learning pipelines—boosting both their performance and convergence rate—it still suffers from identifiable drawbacks, among them a lack of safeguards to verify that parameter updates actually move the policy toward improvement [26].

Beyond this, the TRPO algorithm is itself constrained by certain intrinsic limitations. For one, pinpointing a course of action to lift performance when TRPO training does not produce the hoped-for gains can be a formidable challenge. The practical instantiation of TRPO is constraint-anchored and entails the calculation of the Fisher information matrix, a prerequisite that places a heavy brake on the update tempo.

Multi-agent proximal policy optimization

Spurred by these shortcomings, Schulman *et al.* [27] unveiled the PPO algorithm, built directly on the TRPO lineage. The operational rhythm of PPO alternates between amassing interaction data from the environment and refining a surrogate

© 2023 The Author(s). | Published by Agro Intelligence Research Press

“agent” objective via stochastic gradient ascent. Departing from orthodox policy gradient recipes that apply a single gradient adjustment per data batch, PPO formulates a novel objective that enables minibatch updates across several epochs. The algorithm preserves a subset of TRPO’s beneficial properties while stripping away much of the implementation complexity, delivering broader generality and superior sample economy in applied settings.

Positioned against the specific problem contours and environmental features grappled with in this work, the PPO framework supplies a handful of clear-cut edges over alternative reinforcement learning paradigms: (1) Clipping-Induced Stability: By embedding clipping operations and probability ratio bounds, PPO shackles policy updates to a modest range. Such confinement dampens runaway gradient oscillations, yielding a training process that is more robust and predictable. (2) Broad Network Interoperability: PPO functions harmoniously alongside a wide assortment of neural architectures and function approximator classes, deep networks included. This breadth of compatibility lends it greater flexibility across varied task landscapes and use cases, clearing the path for easier knob-turning of both environmental settings and algorithmic hyperparameters in truck–drone logistics investigations. (3) Fluency with Continuous and High-Dimensional Domains: PPO demonstrates a natural fit for contending with challenges set in continuous action regimes and state manifolds of high dimensionality. This capacity proves especially consequential when disentangling questions related to drone flight path optimization.

These beneficial characteristics position PPO as a highly attractive tool for tackling difficult reinforcement learning problems. Yet, within the present research—which involves numerous agents coexisting in a delivery landscape—the stock PPO algorithm, despite being extensible, encounters challenges when the task demands handling multi-agent collaboration or adversarial dynamics. The difficulty sharpens when the scenario involves a fleet of drones sharing the same operating airspace. It is for this reason that the present effort turns to the MAPPO algorithm, a formulation purpose-built for multi-agent contexts in which ensembles of agents are compelled to reach coordinated choices. As laid out in **Algorithm 1**, MAPPO weaves in the multi-agent abstraction to more nimbly navigate cooperative decision-making currents and the reciprocal influences flowing among multiple intelligent actors.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Initialize critic $Q_{\varphi}u$ and actor π^u with $\theta^u, \forall u$
Initialize the current policies $\pi_{old}{}^u$ with $\theta_{old}{}^u \leftarrow \theta_u$, and the target critic $Q_{\overline{w}u}$ with $\overline{w}^u \leftarrow w^u$
Initialize a memory buffer D
for iteration = 1, 2, . . . , L **do**
s_1 = initiate state
for an episode $t = 1, 2, \ldots, T$ **do**
Each agent u takes action according to $\pi^u_{old}(a^u_t | z^u_t)$
Get the reward r_t, and the next state s_{t+1}
end for
Obtain a trajectory for each UE u : $\tau^u = \{z^u_t, a^u_t, r_t\}^T_{t=1}$
Compute $\{\hat{Q}^u(s_t, \mathbf{a}_t)\}$
Compute advabtages $\{A^u(s_t, \mathbf{a}_t)\}$
Into data buffer D
for mini-batch $k = 1, \ldots, K$ **do**
for each-policy $i = 1, \ldots, n$ **do**
$b \leftarrow$ random mini batch from D for policy i
for each data chunk c in the mini batch b **do**
Update the RNN hidden states for each timestep in c from the first hidden state
end for
end for
end for
end for

Algorithm 1. Multi-Agent PPO

Case description

The experimental runs underpinning this research were coded in Python, drawing on the Gym ecosystem and the MPE toolkit as foundational infrastructure. By applying targeted reshaping to the case environment, the study reduced the multi-agent problem to a single-agent problem, thereby easing the burden of algorithmic testing.

With genuine drone delivery operational realities kept firmly in view, performance yardsticks were constructed by referencing regulatory criteria issued by the Civil Aviation Administration of China (CAAC) and the Federal Aviation Administration of the United States (FAA). These yardsticks were populated with data harvested from both a flight simulation environment called AirSim and logs of actual airborne missions. The study further accounted for drone-specific flight behavior idiosyncrasies, incorporated obstruction data into the model, and assumed prior access to customer positional coordinates to feed the route-planning phase [28].

Extending the scope, the investigation scrutinized the bearing that alternative drone fleet configurations exert on delivery performance and environmental complexity, all while respecting the cooperative coupling between ground vehicles and airborne craft. Presented below is a high-level map of the reinforcement learning environment configuration adopted in this study, cross-referenced against the case framework:

Environment

The present manuscript drew on Gym and MPE to build the reinforcement learning testbed. These two utilities sit within OpenAI's reinforcement learning development stack and slot in comfortably with a range of numerical processing libraries

© 2023 The Author(s). | Published by Agro Intelligence Research Press

and neural network construction kits, including PyTorch and TensorFlow.

Gym serves as a widely used library for quickly spinning up general-purpose reinforcement learning environments. In contrast, MPE operates as a prominent construction kit, purpose-designed for prototyping and evaluating multi-agent environments. In this testbed, factoring in the parameter sets of both aerial drones and ground trucks, the study equipped drones with a spacious action manifold so they could streak toward designated target coordinates with minimal delay. The environment simultaneously encoded obstacle layouts and collision contingencies.

Turning to the customer instantiation mechanism, the study adopted a client-scattering model inspired by principles of Geographic Information Systems (GIS). Individual customers were distributed around discrete stopping nodes, with their spatial offsets from the nodal centroid following a normal distribution. This modeling choice allows the multi-vehicle-plus-drone delivery architecture to sit more congruently with genuine commercial deployment patterns.

State

What gets packed into a state definition can vary markedly from one application context to the next. Proceeding from the distinctive features of the truck–drone logistics model and the focus adopted here, the present article draws a sharp distinction between how states are conceptualized for truck itinerary design and for multi-drone airborne servicing.

Within the truck itinerary design subproblem, the timeline is sliced into discrete intervals that double as states. The atomic state encodes the truck's transition between graph vertices, and a Monte Carlo tree is deployed as the exploratory policy for a solitary vehicle. In contrast, the multi-drone airborne servicing subproblem operates with continuous state variables. These states are assembled from drone and customer spatial coordinates and elapsed time. The internal makeup of the state vector constitutes a meaningful design axis. This study treats entire stopping locations as coarse-grained state categories while employing Euclidean coordinate pairs to pinpoint the spatial whereabouts of trucks, drones, and customers alike. These coordinate pairs are packed into the state representation to streamline distance computations between any two points.

$$S_V = (P_t, r, t) \quad (12)$$

The drone fleet size emerges as a decisive parameter within the truck–drone delivery formulation. The present study treats the drone count as a dial that can be freely turned and proceeds to examine and juxtapose the effects of varying drone numbers under an array of scenario conditions. Additionally, the MAPPO algorithm is used to achieve more consistent behavioral outcomes in drone flight operations. A state S is formalized as

$$S_U = ((x_{c1}, y_{c1}), (x_{c2}, y_{c2}), \ldots, (x_C, y_C), (x_{d1}, y_{d1}), (x_{d2}, y_{d2}), \ldots, (x_D, y_D), r, t) \quad (13)$$

The specification itemizes the elemental building blocks making up a state inside the environment, where (xc,) encodes the spatial coordinates of customer destinations; C captures the total customer cardinality as reflected in the mathematical model and environmental parameters; (xd,yd) encodes the spatial coordinates of other drone positions; D captures the total drone cardinality present in the environment; s records the accumulated score; t measures the passage of time.

Under this state-encoding scheme, the agents—trucks or drones—retain visibility into their counterparts' positions, enabling them to avoid collisions while converging on customer positions in a time-efficient manner. A state representation that incorporates both client locations and delivery vehicle positions more faithfully reflects operational delivery conditions.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Action

The decision space open to a truck agent boils down to choosing which stopping point to focus on next. The action menu is catalogued as follows:

$$A_V = \{P_1, P_2, \ldots, P_p\} \quad (14)$$

Turning to the drone collective, the study draws on the Multi-Agent Particle Environment (MPE) framework, in which the flight corridors traced by multiple drones are treated as continuous-action signals. Additionally, potential drone-obstacle strikes and drone-on-drone aerial impacts are incorporated into the action outcome model. The action menu assigned to each drone agent reads as follows:

$$A_U = \{Froward, Backward, Left, Right\} \quad (15)$$

These paired action frameworks are purpose-fitted to tackle the distinct operational demands of ground-based truck deliveries and air-based drone deliveries within the study's experimental setting.

Reward

Across the span of this investigation, the payoff architecture is pieced together as follows:

Positive Payoffs: Drones receive positive payoffs for completing delivery assignments successfully. These payoffs flow from a base magnitude scaled by a coefficient drawn from the count of clients served within the surrounding zone, itself governed by a normal distribution. It is worth emphasizing that, in accordance with standard delivery protocol, only the drone that first reaches the customer's coordinates is eligible to claim the payoff. To ease the sting of sparse reward regimes—which carry the risk of pushing agents to shun exploratory behavior—drones are handed modest incremental payoffs whenever they shrink the distance separating them from customer coordinates during the course of their flight arcs. This design element prods drones toward speedier delivery completion.

Negative Payoffs: Penalty payoffs are levied under specified circumstances, including redundant delivery attempts, aerial collisions, and contact with obstacles. These penalty signals are designed to steer drones away from becoming mired in such predicaments. The study undertakes active adjustment and delicate recalibration of the payoff settings as the learning process unfolds. As shown in **Figure 4a**, during the early experimental phase, the collision penalty magnitudes were calibrated too aggressively, leading drones to prioritize mutual avoidance above all else. The upshot was that the early payoff traces churned out values skewed heavily toward the negative.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

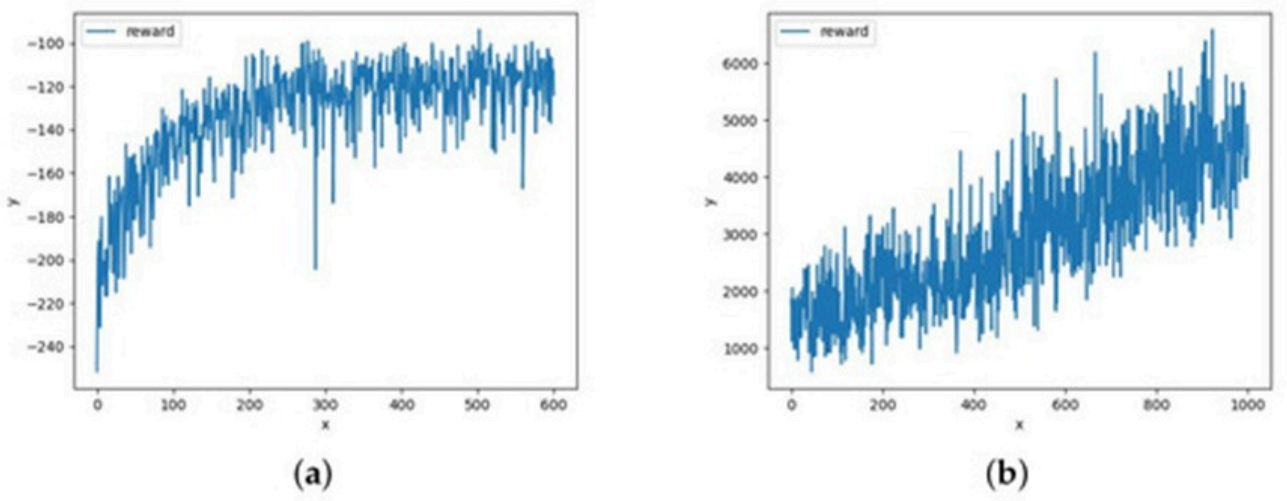

Figure 4. MAPPO reward convergence curve: (a) MAPPO-Drone collision penalty is high, and (b) MAPPO-After collision penalty adjustment.

Payoff Appraisal: Casting the truck–drone cooperative fulfillment problem as a fully collaborative game under a game-theoretic optic, a Critic network is used to evaluate the ensemble of actions and funnel reward processing through a centralized conduit. This design choice carves out a path for the study to gauge the incremental value added by individual drones to the broader delivery operation. The study also normalizes each reward stream by subtracting the mean and dividing by the standard deviation. This normalization step aligns the reward signal distribution with a standard normal distribution, with mean zero and variance one. The reward normalization device compresses the timeline required for learning and convergence. In multi-agent environments, reward streams typically exhibit pronounced choppiness, a quality that can easily destabilize training or lead to outright divergence. Reward normalization helps flatten out this choppiness, thereby improving the stability of the training run.

Experiment and Results

What follows documents the testing regimen applied to the Gym environment engineered during this investigation. We kicked off by contrasting the learning trajectory and terminal performance observed with and without the imposition of action-space restrictions. Following that, a suite of experiments was executed within this model to probe how distinct parameter selections shape eventual outcomes. We wrapped up by gauging the robustness of both the environmental scaffold and the algorithmic machinery, running side-by-side comparisons with the MAPPO algorithm across various parameter calibrations to map its performance envelope.

Every experimental cycle shared a single, fixed environment foundation. The hardware underpinnings consisted of an AMD Ryzen 7 5800H CPU at 3.2 GHz, 16 GB of DDR4 RAM at 3200 MHz, and an RTX 3060 with 8 GB of VRAM. The software layer comprised Python 3.7.12, Pytorch 1.5.0, and Gym 0.10.5.

The analytical lens of this work was trained on the truck–drone routing challenge as instantiated through the MPE toolkit. We pursued a line of inquiry to tease apart the effects of collision penalty strengths, drone fleet cardinality, and customer pool size. A foundational scenario featuring two trucks in cooperative operation served as the control case, from which we fanned out into conditions labeled sparse drone deployments (fewer than 5), moderate drone deployments (5–12), and dense drone deployments (exceeding 12). Customer allotments were scaled relative to the drone count, with drone flight endurance serving as a binding constraint. The floor for customer numbers was set equal to the drone count, and the ceiling was bounded at three times the drone count, a design choice intended to more closely approximate real operational landscapes.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Regarding environmental architecture, the truck's routing logic was modeled as a discrete selection task, while the drone's aerial maneuvering problem was framed as a continuous control task. Partitioning the problem along these lines yielded a meaningful compression of the action space, carving a tractable route toward solving the truck–drone challenge at separate layers of abstraction. In the concrete instantiation, two trucks launched from assigned starting coordinates, swept across all designated stopping nodes, and ultimately terminated at the final waypoint. The drone operating domain incorporated stochastic elements, whereas customer placements drew on a known prior. Drones were issued penalty signals whenever their flight paths strayed into hazardous proximity to obstacles or to one another.

As for drone flight specifications and hardware assumptions, the study was firmly rooted in the metropolitan realities of truck–drone logistics. Given that a sizable fraction of city administrations prohibit heavy truck traffic through urban cores, this work restricted its purview to medium-duty trucks acting as drone carriers. The pronounced disparity in physical scale between drones of different payload brackets was also accounted for. The study's exemplar set included a 2 kg payload drone, currently undergoing trials for experimental food conveyance, alongside a 20 kg payload drone, the type deployed by certain maritime logistics operators for expedited vessel-to-shore transfers. At this juncture, however, we are not investigating heterogeneous fleets blending multiple drone categories. Within these boundaries, the experimental lower limit for drone count was anchored at 4, and the upper limit was fixed at 18.

Operating within the algorithmic constraints and environmental specifications set by this study, early-phase drones could scarcely avoid colliding with one another. Accordingly, we softened the penalty for close-quarters drone encounters and boosted the reward for successfully reaching delivery waypoints. The graphic below displays reward traces captured from the MAPPO algorithm across a spread of parameter configurations:

As shown in **Figure 4**, tinkering with the penalty for drone proximity materially improves the likelihood of stronger rewards upon target interception. The salient distinction between **Figure 4a** and **Figure 4b** lies in whether the sanction for drones entering one another's vicinity has been rescinded. In **Figure 3b**, with a configuration of 15 drones, 45 customers, and 6 barrier structures, the MAPPO algorithm was run for 5 million steps. The conclusion drawn was that MAPPO exhibits a reliable capacity to inch toward the near-optimal solution frontier, with the peak reward brushing roughly 8000 after the initial penalty magnitudes had been tapered.

A close reading of the per-episode reward traces, viewed through the side-by-side lens of the figure above, makes it unmistakably clear that MAPPO achieves rapid stabilization in trials with small drone cohorts. Although MAPPO appears to struggle with convergence when the drone count balloons, this behavior stems from the exponential increase in the informational load agents must shoulder in densely populated drone environments, which slows learning.

Based on **Table 2** and the reward traces presented above, MAPPO clearly establishes a competitive edge over alternative algorithmic approaches. Stacking it against the MADDPG algorithm and its enhanced variant, the Twin Delayed MADDPG (MATD3) algorithm, it becomes evident that under small-scale test conditions, both MADDPG and MATD3 converge more slowly than MAPPO, while plateauing at lower reward ceilings.

Table 2. Algorithm result comparison.

Algorithm scale	Mean reward	Steps per episode	Episodes	Obstacle count	Customer count	Drones count
MAPPO-L	1372.61	1000	100	6	45	15
MAPPO-M	173.24	200	100	4	12	8
MAPPO-S	68.96	100	100	3	6	4
MATD3-L	1134.82	1000	100	6	45	15
MATD3-M	168.19	200	100	4	12	8

© 2023 The Author(s). | Published by Agro Intelligence Research Press

MATD3-S	56.83	100	100	3	6	4
MADDPG-L	965.71	1000	100	6	45	15
MADDPG-M	121.53	200	100	4	12	8
MADDPG-S	37.52	100	100	3	6	4

Scale: L = Large, M = Medium, and S = Small.

As **Figure 5** underscores, a collection of box plots, sourced from single-run results picked at random, spotlights the commanding edge of the MAPPO algorithm when juxtaposed with MATD3 and MADDPG. Tellingly, even inside the medium-scale evaluation regime depicted in **Figure 5b**, where the central tendency of MAPPO modestly trails that of MATD3, MAPPO nonetheless manages to nose ahead of MATD3 when it comes to the uppermost achieved values. What stands out most is that MAPPO consistently delivers substantially narrower dispersion in outcomes than its two algorithmic rivals.

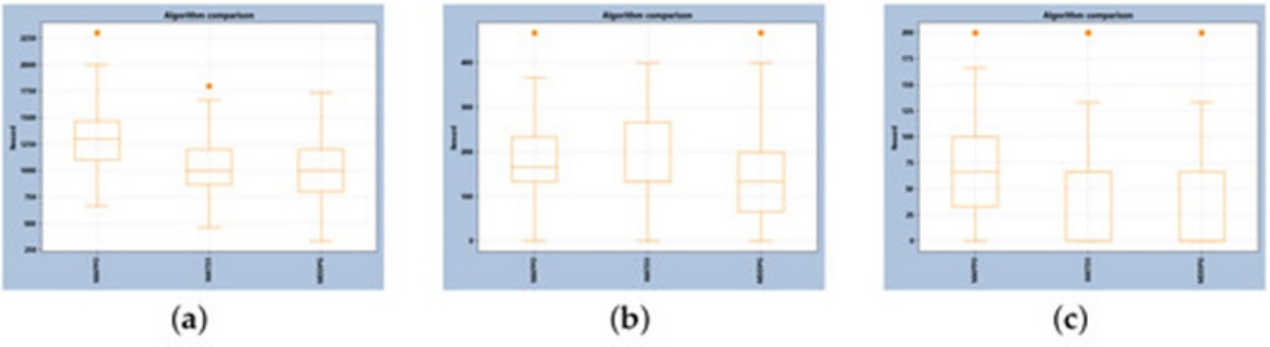

Figure 5. Result of a single run: (a) Test-L, (b) Test-M, and (c) Test-S.

Homing in further, as shown in **Figure 6**, the reward traces clearly indicate that the MAPPO algorithm follows a sharper, less jittery upward trajectory in the study's custom environment. The computational throughput metrics bear this out: MAPPO describes a comparatively seamless climb, whereas the MADDPG algorithm not only drains more wall-clock time but also produces reward curves that oscillate far more wildly than those generated by MAPPO. Additionally, the MADDPG algorithm places a greater memory burden than MAPPO.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

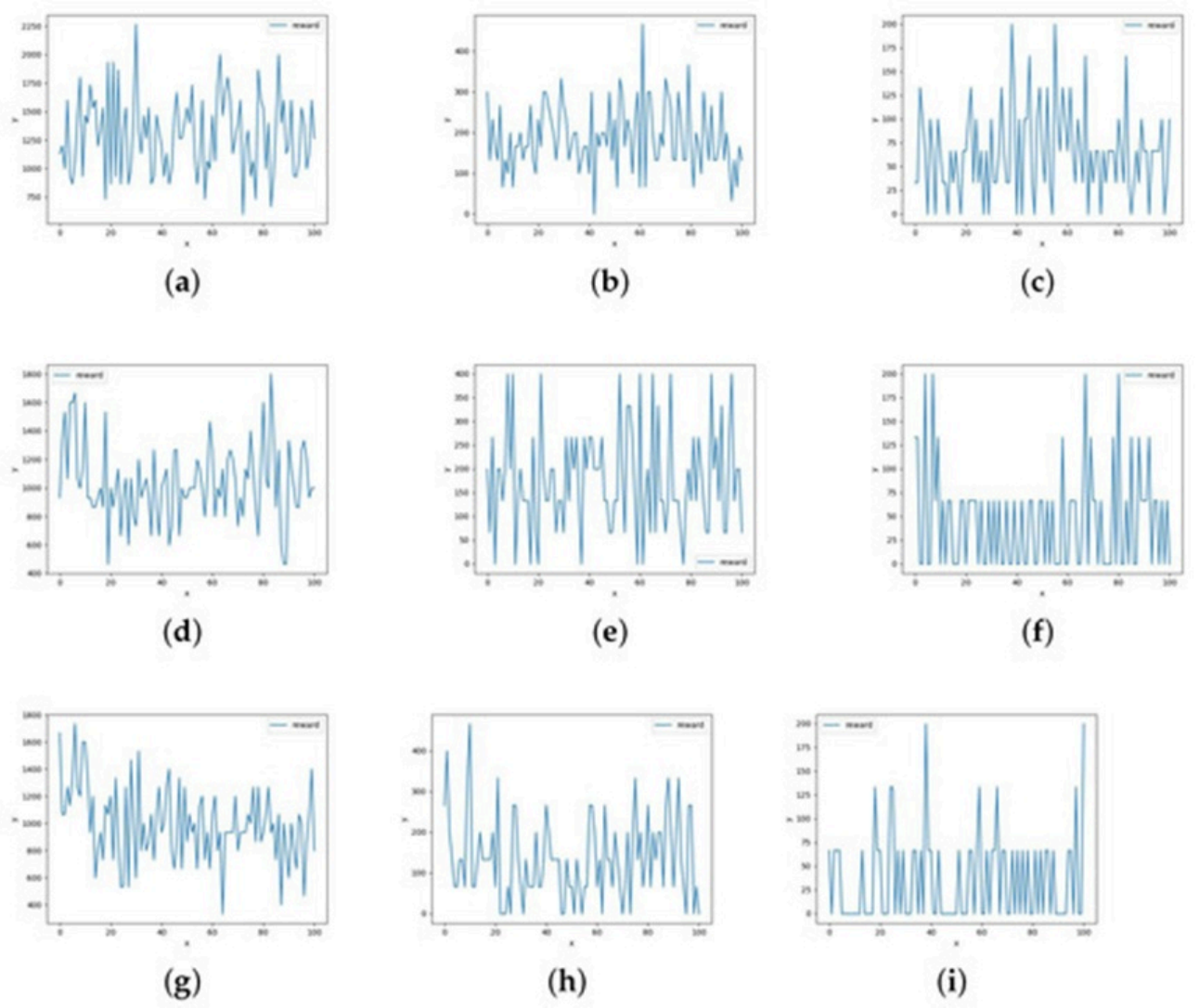

Figure 6. Algorithm reward graph: (a) MAPPO-Large case, (b) MAPPO-Medium case, (c) MAPPO-Small case, (d) MTD3-Large case, (e) MTD3-Medium case, (f) MTD3-Small case, (g) MADDPG-Large case, (h) MADDPG-Medium case, and (i) MADDPG-Small case.

From the weight of the evidence, one can safely infer that the MAPPO algorithm holds the upper hand along the fronts of temporal frugality, memory parsimony, and solution fidelity when tasked with cracking the multi-agent truck–drone delivery puzzle. MAPPO stands ready to unlock superior outcomes inside the multi-agent theater of truck–drone logistics.

Pulling the threads together, the experimental harvest reaped from this case study affirms that fielding the MAPPO algorithm for intelligent vehicle–drone dispatch yields considerable dividends. Beyond this, tuning parameters and adopting techniques such as reward normalization can further improve the algorithm's learning rate and terminal competence. A diverse array of experimental runs, covering a range of agent population sizes and parameter settings, was executed. The insights distilled from these experimental forays carry meaningful weight for the ongoing evolution and polishing of vehicle–drone intelligent dispatch architectures [29].

Conclusion

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Across the pages of this paper, we locked horns with the truck–drone delivery optimization challenge through the prism of multi-agent reinforcement learning. We leveraged the MPE framework alongside the MAPPO algorithm to dissect this tangled problem, balancing truck route design with drone flight corridor planning under a bouquet of competing objectives. Capitalizing on the structural hallmarks of truck–drone logistics, we opted for a tiered solution methodology, pressing Monte Carlo Tree Search (MCTS) and reward normalization into service to shrink training wall time and tighten solution convergence. This strategy did not merely quicken the tempo of problem resolution; it also stretched the repertoire of methods applicable to vehicle–drone fulfillment.

Of particular note, at the crossroads where multi-agent systems meet aerial robotics, we surfaced consequential insights through a sustained campaign of comparative experimentation. We dissected the behavioral signature of the MAPPO algorithm across shifting parameter landscapes, charting its influence on both training outcomes and training cadence. After cycling through a large volume of trials, we settled on the practical guideline that, for real-world vehicle–drone delivery rollouts, it pays to trim the number of customer points assigned to any single drone, steering drones toward a mode where each aerial unit focuses on a solitary customer throughout its flight [30]. We further established that, within the multi-agent domain of vehicle–drone logistics, MAPPO comprehensively outperforms MADDPG across runtime, memory consumption, convergence speed, and solution quality. It follows that sustained inquiry into the diverse performance metrics governing vehicle–drone delivery, paired with the deployment of the multi-agent MAPPO algorithm to drone flight trajectory and service modality problems, represents a richly promising vein of research.

Shifting to the experimental dimension, anchoring reinforcement learning within the MPE ecosystem furnishes a far more malleable proving ground for trialing disparate algorithms and eases the workflow for comparative benchmarking and reward-function tweaking. As e-commerce and the logistics sector surge forward on successive tides of technological advancement, the vehicle–drone delivery paradigm becomes an ever more compelling object of study. Extrapolating from the research outlined in this paper, prospective paths of inquiry and avenues of refinement for vehicle–drone delivery models could include expanding the application of a broader palette of multi-agent algorithms for testing and reward-function tuning. Entertaining the use of high-fidelity simulation platforms along the lines of AirSim, which are erected atop game engines, to recreate authentic operational conditions inside richly detailed three-dimensional virtual environments [10, 31]. Mounting field trials with physical drone hardware to exercise flight control routines or execute point-to-point parcel handoffs, thereby accelerating the migration of drone technology into the logistics mainstream [32, 33].

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
DQN	Deep Q-Network
MLP	Minimum latency problem
TRP	Traveling repair problem
TSP	Traveling Salesman Problem
TRPO	Trust Region Policy Optimization
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MAPPO	Multi-Agent Proximal Policy Optimization
MPE	Multi-Agent Particle Environment

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 01 Jul 2022 Revised: 05 Sep 2022 Accepted: 21 Sep 2022
Published online: 25 November 2023

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Erdelj M, Natalizio E. UAV-assisted disaster management: Applications and open issues. In: Proceedings of the 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 2016; IEEE: Piscataway, NJ, USA, 2016. p. 1–5.

2. Sajid M, Mittal H, Pare S, Prasad M. Routing and scheduling optimization for UAV assisted delivery system: A hybrid approach. Appl Soft Comput. 2022;126:109225.

3. Cheng N, Wu S, Wang X, Yin Z, Li C, Chen W, et al. AI for uav-assisted iot applications: A comprehensive review. IEEE Internet Things J. 2023;10:14438-61.

4. Yang M, Liu G, Zhou Z, Wang J. Partially observable mean field multi-agent reinforcement learning based on graph attention network for uav swarms. Drones. 2023;7:476.

5. Gu J, Wang J, Guo X, Liu G, Qin S, Bi Z. A metaverse-based teaching building evacuation training system with deep reinforcement learning. IEEE Trans Syst Man Cybern Syst. 2023;53:2209-19.

6. Bi Z, Guo X, Wang J, Qin S, Liu G. Deep reinforcement learning for truck-drone delivery problem. Drones. 2023;7:445.

7. Jung S, Kim J. Adaptive and stabilized real-time super-resolution control for UAV-assisted smart harbor surveillance platforms. J Real-Time Image Process. 2021;18:1815-25.

8. Hu B, Wang J. Deep learning based hand gesture recognition and UAV flight controls. Int J Autom Comput. 2020;17:17-29.

9. Wang C, Lan H, Saldanha-da Gama F, Chen Y. On optimizing a multi-mode last-mile parcel delivery system with vans, truck and drone. Electronics. 2021;10:2510.

10. Feng Z, Huang M, Wu D, Wu EQ, Yuen C. Multi-Agent reinforcement learning with policy clipping and average evaluation for uav-assisted communication markov game. IEEE Trans Intell Transp Syst. 2023;24:14281-93.

11. Liu Z, Li X, Khojandi A. The flying sidekick traveling salesman problem with stochastic travel time: A reinforcement learning approach. Transp Res Part E Logist Transp Rev. 2022;164:102816.

12. Dai M, Huang N, Wu Y, Gao J, Su Z. Unmanned-Aerial-Vehicle-Assisted Wireless Networks: Advancements, Challenges, and Solutions. IEEE Internet Things J. 2022;10:4117-47.

13. Xu G, Zhang Y, Ji S, Cheng Y, Tian Y. Research on computer vision-based for UAV autonomous landing on a ship. Pattern Recognit Lett. 2009;30:600-5.

14. Meng Y, Wang W, Han H, Ban J. A visual/inertial integrated landing guidance method for UAV landing on the ship. Aerosp Sci Technol. 2019;85:474-80.

15. Wang J, Li D. Resource oriented workflow nets and workflow resource requirement analysis. Int J Softw Eng Knowl Eng. 2013;23:677-93.

16. Wang Y, Gao Z, Zhang J, Cao X, Zheng D, Gao Y, et al. Trajectory design for uav-based internet of things data collection: A deep reinforcement learning approach. IEEE Internet Things J. 2022;9:3899-912.

17. Freitas JC, Penna PHV, Toffolo TA. Exact and heuristic approaches to truck-drone delivery problems. EURO J Transp Logist. 2022;12:100094.

18. Chang YS, Lee HJ. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst Appl. 2018;104:307-17.

19. Kitjacharoenchai P, Ventresca M, Moshref-Javadi M, Lee S, Tanchoco JM, Brunese PA. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. Comput Ind Eng. 2019;129:14-30.

20. Moshref-Javadi M, Lee S, Winkenbach M. Design and evaluation of a multi-trip delivery model with truck and drones. Transp Res Part E Logist Transp Rev. 2020;136:101887.

21. Jeong HY, Song BD, Lee S. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. Int J Prod Econ. 2019;214:220-33.

22. Goyal A, Kumar N, Dua A, Kumar N, Rodrigues JJPC, Jayakody DNK. An Efficient Scheme for Path Planning in Internet of Drones. In: Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM); 2019; Big Island, HI, USA; 2019. p. 1-7.

23. Poikonen S, Golden B, Wasil EA. A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS J Comput. 2019;31:335-46.

24. Tang Z, Van Hoeve WJ, Shaw P. A study on the traveling salesman problem with a drone. In: Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Proceedings of the 16th International Conference, CPAIOR 2019; 2019; Thessaloniki, Greece. Proceedings 16. Berlin/Heidelberg, Germany: Springer; 2019. p. 557-64.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

25. Barto AG, Sutton RS, Anderson CW. Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Trans Syst Man Cybern. 1983;SMC-13:834-46.

26. Engstrom L, Ilyas A, Santurkar S, Tsipras D, Janoos F, Rudolph L, et al. Implementation matters in deep rl: A case study on PPO and TRPO. In: Proceedings of the International Conference on Learning Representations; New Orleans, LA, USA; 2019.

27. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv. 2017;arXiv:1707.06347.

28. Mbiadou Saleu RG, Deroussi L, Feillet D, Grangeon N, Quilliot A. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. Networks. 2018;72:459-74.

29. Canese L, Cardarilli GC, Di Nunzio L, Fazzolari R, Giardino D, Re M, et al. Multi-agent reinforcement learning: A review of challenges and applications. Appl Sci. 2021;11:4948.

30. Hammami SE, Afifi H, Moungla H, Kamel A. Drone-assisted cellular networks: A multi-agent reinforcement learning approach. In: Proceedings of the ICC 2019 IEEE International Conference on Communications (ICC); 2019; Shanghai, China. Piscataway, NJ, USA: IEEE; 2019. p. 1-6.

31. Xue Z, Gonsalves T. Vision based drone obstacle avoidance by deep reinforcement learning. AI. 2021;2:366-80.

32. Cui J, Liu Y, Nallanathan A. Multi-agent reinforcement learning-based resource allocation for UAV networks. IEEE Trans Wirel Commun. 2019;19:729-43.

33. Song Y, Romero A, Müller M, Koltun V, Scaramuzza D. Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. Sci Robot. 2023;8:eadg1462.

© 2023 The Author(s). | Published by Agro Intelligence Research Press