Environmentally-Aware Robotic Vehicle Networks Routing Computation for Last-mile Deliveries

Chengyi Qu*, Rounak Singh*, Sharan Srinivas† and Prasad Calyam†
*† Department of Electrical Engineering and Computer Science, University of Missouri - Columbia, USA.
Email: *{cqy78, rsft6}@mail.missouri.edu, †{srinivassh,calyamp}@missouri.edu

***Abstract*—For next-generation logistics management, robotic vehicles such as autonomous ground robots and aerial drones can alleviate the strain on last-mile distribution. They can help avoid on-road congestion, navigate hard-to-reach locations, and parallelize delivery operations. However, as the robotic vehicles move in a given delivery area, environmental barriers e.g., trees or buildings, affect air-to-air (A2A), air-to-ground (A2G), ground-to-ground (G2G) network communications on a hybrid truck-drone-robot system. In this paper, we present an environmentally-aware cooperative network routing computation scheme to avoid obstacle blockage in A2A/A2G/G2G network communications for addressing large-scale coordinated operations of the hybrid truck-drone-robot system. Specifically, we propose an offline policy-based routing algorithm and two online extensions (i.e., heuristics and learning-based) to solve the hybrid last-mile delivery vehicles communication problem in order to trade-off between end-to-end communication (i.e., increase network throughput) and delivery efficiencies (i.e., lower parcel delivery time consumption). We evaluate our scheme using state-of-the-art network routing algorithms in a trace-based simulator that integrates both the vehicles and networking sides. Performance evaluation results from our simulations show that: (i) our offline approach is Pareto-optimal among non-learning supported algorithms in a pre-delivery scenario, and (ii) our RL-based online algorithm achieves between 85-96% of the Oracle strategy performance during delivery procedures.**

***Index Terms*—Robotic vehicle networks, Last-mile parcel delivery, Reliable network communication, Learning-based scheduling**

I. Introduction

The last-mile delivery market is projected to have a compounded annual growth rate (CAGR) of 16% during the next five years, primarily due to the proliferation of e-commerce [1]. In addition, the last-mile delivery cost, which already accounts for 41% of the total supply chain cost, is also expected to increase [2]. Adopting emerging logistics technologies, namely autonomous ground robots and aerial drones, holds great promise for alleviating the strain on last-mile distribution. Specifically, they can avoid on-road congestion, navigate hard-to-reach locations, and parallelize delivery operations. Moreover, major corporations such as Google, Amazon, and others have developed technologies for bringing the potential of a single type of autonomous robotic vehicles to the field of last-mile parcel delivery [3]. However, there is a dearth of methods that address large-scale coordinated operations of *a hybrid truck-drone-robot system.*

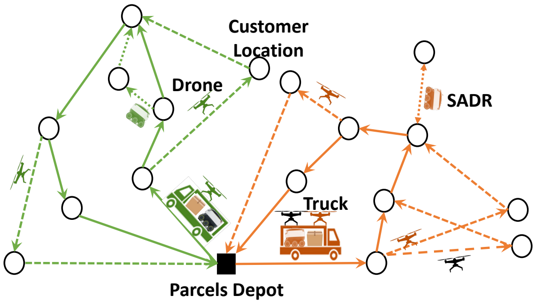


Fig. 1. Map of a hybrid truck-drone-robot last-mile parcel delivery scenario with distributed customers in a random area, and multiple trucks carrying different number of drones and SADRs to deliver parcels to customers.

Figure 1 shows an example of hybrid truck-drone-robot system with distributed customers in a complex urban area (i.e., the delivery application scenario). In this system, we assume that there is a collaboration among three vehicle types - traditional trucks, sidewalk autonomous delivery robots (SADRs), and autonomous unmanned aerial vehicles (UAVs) or drones. The map corresponds to a customer delivery center in a random area, where the depot, trucks, robotic vehicles and customer locations are identified. In this delivery application scenario, one depot interacts with multiple trucks that are communicating at the same time with different robotic vehicles (i.e., SADRs and drones). These robotic vehicles are tasked with delivery of parcels to various last-mile customers/homes that are in geographically distributed areas.

The primary motivation for the use of the hybrid truck-drone-robot system is that the limitations of one vehicle type are complemented by the others. Drones leverage the low-altitude airspace to avoid traffic congestion and travel faster than the other two vehicle types, but have limited payload capacity and flying range [4], [5]. The SADRs can efficiently transport multiple heavier parcels, operate under extreme weather conditions, and access areas that make drone deliveries difficult or inefficient, but are restricted in terms of speed, road network, and battery range [6]. On the other hand, trucks can overcome the limitations of drones and SADRs with respect to payload capacity, and range. Nevertheless, they cannot bypass traffic congestion and spend a lot of time searching for a parking space, especially in congested urban areas. Besides, trucks are emission-heavy and are not as environmentally friendly as drones or rover robots [7]. Thus, a hybrid truck-drone-robot system that works in tandem has the potential to exploit the individual strengths of drones and SADRs and achieve better last-mile delivery efficiency.

In the above delivery application scenario, robotic vehicles can be assumed to be connected together in robotic vehicle networks instantiated by a ground control station (GCS). This network can be used to track the delivery states for robotic vehicles [8], dynamically reorganize the delivery tasks [9] and prevent robotic vehicles from getting stolen or hacked [10]. However, obstacles in the flight path influence the air-to-air (A2A), the air-to-ground (A2G) and the ground-to-ground (G2G) network communications between drones, SADRs and the GCS, respectively. The obstacles could involve physical environmental barriers such as trees, buildings or other man-made constructions. Moreover, obstacles could also be caused by weather barriers such as wind strength, direction and even air humidity. The robotic vehicle network needs to have resilient network connectivity via a highly adaptive path computation scheme. A resilient robotic vehicle network allows a warehouse company to quickly act on decisions based on situational awareness, and allows sharing of information that is useful to perform on-the-fly reroutes of the drone paths. For instance, a quick reroute decision from the GCS can be useful during irregular weather conditions (e.g., on a windy day) that significantly influence the drones' flight direction as well as flight time during scheduled deliveries in general. State-of-the-art in environmental awareness for path computation solving can be seen in the work on [11] and [12]. None of the existing works solve the delivery application problem with environmental awareness, while also optimizing the delivery efficiency as well as handling environment obstacles that can lead to reduced performance and/or failure of network communications in a hybrid truck-drone-robot system.

In this paper, we address the above important knowledge gap by introducing a novel *ENvironmentally-aware COoperative robotic vehicles' Routing nEtwork computation scheme, viz. ENCORE* to solve the hybrid last-mile delivery vehicles communication problem in order to trade-off between end-to-end communication (i.e., increase network throughput) and delivery efficiencies (i.e., lower parcel delivery time consumption). First, our *ENCORE-offline* approach builds upon the geographic routing scheme, and involves proactively ensuring parcel delivery efficiency, while maximizing the throughput of the end-to-end communication during the delivery procedure. Our *ENCORE-offline* approach provides resilience to robotic vehicle networks that are relevant to the delivery application scenario by overcoming barriers due to obstacles that impact A2A and A2G network communications. The resilience feature in *ENCORE-offline* geographic routing is due to our ability to ensure that the reconstruction of the A2A, A2G or G2G network communications has limited overhead on the Oracle delivery task, which is attributed to the state-of-the-art parcel delivery scheduling algorithm. In order to accommodate highly adaptive path computation cases in real-time, we propose two online approaches to enhance the *ENCORE-offline* approach: (i) *Heuristic-based (ENCORE-Heu)*, and (ii) *Learning-based (ENCORE-RL)*. These enhancements improve the effectiveness (i.e., shorten the unit re-computation time) of the *ENCORE-offline* approach and increase the performance gain in throughput per unit delivery task. The heuristics-based approach uses the updated Deep Deterministic Policy Gradient (DDPG) [13] algorithm for prediction of the positioning of the robotic vehicles to greedily generate a local optimized path. The learning-based approach is suitable for near-optimally balancing time constraints and providing environmental obstacle awareness using Reinforcement Learning principles [14].

The paper remainder is organized as follows: Section II describes related work. Section III presents our last-mile parcel delivery problem with time and obstacle constraints. Section IV details algorithms to solve the drone path computation problem with our proposed *ENCORE-offline*, as well as the heuristics-based and learning-based online enhancements. We detail performance evaluation results with trace-based simulations in Section V. Section VI concludes the paper.

II. Related Work

The authors in [15] introduced the flying sidekick traveling salesman problem (FSTSP) that addresses the challenge of determining routes for a drone working in tandem with a truck in a last-mile delivery scenario. For this issue, a mixed integer linear program (MILP) formulation was proposed to optimally solve the problem. The same work then investigated the parallel drone scheduling TSP (PDSTSP) variant giving both optimal and heuristic approaches. However, this work does not consider time consumption and obstacle issues that are critical in realistically modeling the interaction between the drone and the environment.

There have been works [16] and [17] that consider the limitation of the robot vehicles' energy consumption but do not account for obstacles, while managing the network for completing a given delivery task. Similarly, the problem of delivering parcels with drones considering optimization of flight time consumption was investigated in [18]. Initially, their approach involved assuming each segment traveled by the robotic vehicles to be subject to an unknown cost in terms of operation time. After a certain number of consecutive missions, their scheme learned better paths from the previous routes, so as to optimize the range in future missions. In contrast to these prior works, our proposed approach for communication path computation considers environmental awareness in terms of time consumption as well as the presence of obstacles (e.g., trees, buildings) in end-to-end computations at the GCS for A2A, A2G and G2G links. The recent work in [19], [20] is closely related to our work in terms of routing multiple robotic vehicles autonomously operating from a truck in order to serve one or more customers, and return to the same truck for a battery swap and package retrieval. Our approach is also motivated by the work in [9], where the authors that first introduced FSTSP extended their work in the form of the multiple FSTSP (mFSTSP), in which a delivery truck and a heterogeneous fleet of drones coordinate to deliver small parcels to many customers. In our hybrid system approach, few SADR nodes are also introduced to enhance the communication link generation as well as shorten the time of the overall parcel delivery procedure.

Routing protocol schemes for robotic vehicle mobility management considering time consumption and obstacles awareness have been widely investigated recently [21]–[23]. In these works, energy consumption issues for robotic devices (e.g., SADRs or drones) management in the context of the multi-access edge computing paradigm are well addressed. For instance, author in [21] considers mobility problems of the robotic vehicles while handling user requirements of time conservation over low-latency or vice versa. Our proposed

ENCORE-offline algorithm builds on these works that represent the most recent advances in the area of geographic routing algorithms that perform better than other stateless geographic routing solutions, as well as stateful mesh routing in terms of packet delivery success ratio and path stretch.

III. The Last-mile Parcel Delivery Problem

In this section, we introduce the urban area parcel delivery problem terminology featuring a hybrid truck-drone-robot system as shown in Figure 1. Following this, we formulate the robotic vehicle path computation problem to handle vehicle operation time and obstacle constraints of parcel delivery using drones and SADRs.

A. Last-mile Parcel Delivery Terminology

Let $G = (V, E)$ be the graph that models the *delivery area* (city), where $V = v_0 \cup V^C$ is the set of vertices represented by the *depot* location v_0 and the possible n *customers* locations $V^C = \{v_1, \dots, v_n\}$, and $E = \{\{v_i, v_j\} | (v_i, v_j) \in V^2, i \neq j\}$ is the set of undirected edges that represents the available connections on the ground (roads) between locations. Notice that, in general, the graph G is not a complete graph, i.e., the number of available edges $|E| \ll \frac{n(n-1)}{2}$. The location of the depot v_0 is centered at the origin of a 3D Cartesian coordinate system in $(0, 0, 0)$, while each vertex $v \in V^C$ is characterized by a triple of relative coordinates (x_v, y_v, z_v) with respect to the depot position. Such a triple, represents the locally converted physical global position retrieved by the GPS, i.e., latitude, longitude, and altitude.

Let $d_{uv} = \|u - v\|$ be the Euclidean distance between vertices $u, v \in V$. For each $(u, v) \in E$, let t^T_{uv} be the time employed by a truck for driving from u to v along the edge (u, v), and let t^D_{uv} be the required time by a robotic vehicle for operating (i.e., autonomous flying or driving) from any u to v. Let s^T be the constant speed of the truck, and let s^D be the constant speed of the robotic vehicle. Each fleet can accomplish only a single global mission. For simplicity, we assume that the weight of the parcels for drone is fixed to w_d, and the weight of the parcels for SADR is fixed to w_s. Hence, in each sub-route, a drone or a SADR carries an additional payload w from the truck to the customer, and returns empty from the customer to the truck. The truck has an unbounded travel limit and we always assume that it can deliver parcels to all the customers without the help of drones. With this assumption, we ensure that the truck is not time-constrained. However, drones and SADRs are time-constrained and they have a fixed amount of operation time T at the beginning of each global mission.

To serve dynamic cooperative drone delivery, we apply the JOCR-U algorithm [20] at the beginning of the delivery task to schedule the initial delivery sequence on each robotic vehicle and within the required time period. The JOCR-U algorithm optimally solves the problem of minimizing the time required to deliver all parcels and return to the depot. It also considers the cost per vehicle used during the delivery procedure. Minimizing total cost and delivery completion time are two conflicting objectives for the problem under study. For instance, if minimizing delivery completion time is the sole objective, the JOCR-U models may utilize all available robotic vehicles to exploit simultaneous order fulfillment. While this may shorten the overall completion time, it increases the fixed cost of using unmanned devices.

In our proposed model, we consider two metrics: (i) communication throughput metric, and (ii) delivery time metric. The communication throughput metric can be positively correlated with the cost metric introduced in JOCR-U for the following reasons: once the distance is long, the environment will become complicated (i.e., the number of obstacles will increase), and the robotic vehicle is likely to move out of the original ground equipment (i.e., warehouse and truck) communication range. A mesh network is needed to maintain the connection to confirm the delivery and inform the robotic vehicles about where to return back, which will require the *ENCORE-offline* algorithm to establish a communication route. Therefore, the main goal of the proposed last mile package delivery is to optimize the trade-off between A2A, A2G and G2G communication link performance and the overall delivery time.

B. Robotic Vehicle Routing Problem Formulation

Let us consider the following model with node n (e.g., drone u or SADR s) forwarding packet p towards destination d. The purpose of the network path computation is to reconstruct the connection between A2A, A2G and G2G communication links, and create a communication path between robotic vehicle and the destination truck eventually. To this aim, in this model, node n needs to decide which neighbor should receive updated packet p to progress towards d while balancing between the maximum delivery completion time for the whole procedure, and the total throughput of p w.r.t obstacle conditions. Once robotic vehicle starts a delivery task, it requires a communication link generation and the total throughput is not promising due to obstacle issue and limited communication range. On the other hand, if all delivery tasks are completed by the truck itself, it promises to communicate with maximum throughput. However, truck transportation alone will waste a lot of time in comparison due to: (i) complex environmental conditions in urban areas (for example, traffic problems, labor), and (ii) rectilinear distance movement trajectories. We formally summarize the above with the following formula:

$$f(d.x, d.y, d.z, \theta, n) = \theta \cdot \|\delta(n, d)\| + (1 - \theta) \cdot \|\tau\| \qquad (1)$$

where, $\delta(n, d)$ is the convex potential updated time of node n with respect to the destination node d that allows packet transmission with approximation of the shortest path by reapplying JOCR-U on route n to d. τ is the total calculated maximum delivery completion time at node n. θ is a parameter to balance between the delivery time τ and the updated shortest path approximation as well as the updated time. The purpose of the path computation problem is to find the most appropriate θ value which could save the most delivery time overall while also keeping the network connection continuous and stable.

In order to compute $\delta(n, d)$, the GCS on truck needs geographic information about physical obstacles that can potentially cause packet drops due to lack of wireless coverage near their geographical locations. We assume that when an obstacle blocks the communication link, it is described as a long elliptical space between the robotic vehicle node and the corresponding GCS node antenna. The center of the circle on all sections of the ellipsoidal region falls on the line connecting the transmitter and the receiver. The equation to calculate the Fresnel zone radius of each section at the boundary is given as follows [24]:

$$F = \sqrt{\frac{\lambda \times d_1 \times d_2}{d_1 + d_2}} \tag{2}$$

where, F is the Fresnel zone radius, d_1 and d_2 are the distances from section boundary to both ends, and λ is the wavelength of the radio signal. In order to ensure the quality of the communication, the recommended blockage of the obstacles is up to 20%. If the intrusion of obstacles exceeds the 20% of the Fresnel zone, we consider it as a communication block between the robotic vehicles and the corresponding ground nodes. In this case, we need to recalculate the path to find out the alternative route from the path, which requires the nodes to be aware about the obstacle location and nearby neighbours (i.e., other air or ground robotic vehicles).

Once node n is aware of its new propagation radius R_{obs} and the center coordinates $C_{obs}.x$, $C_{obs}.y$ and $C_{obs}.z$ of the obstacle, it computes $\tau(n,d)$ as follows:

$$\delta(n,d) = \sum_{i=1}^{M} \frac{O_i(d.x,d.y,d.z) - 1/\sqrt[\alpha]{dist}}{dist(n.x,n.y,n.z,C_i.x,C_i.y,C_i.z)^{\alpha}} \tag{3}$$

where, $dist(x_{1,2}, y_{1,2}, z_{1,2})$ is the geographical distance; α is the attenuation order of obstacles' potential field that has been shown to provide potential soft optimal performance. Specifically, $O_i(d.x,d.y,d.z)$ is calculated given the obstacle i, the intensity induced by the destination node d:

$$o_i = \frac{R_i^{\alpha}}{\alpha \cdot (dist(d.x,d.y,d.z,C_i.x,C_i.y,C_i.z) + R_i)^2} \tag{4}$$

In order to calculate τ, both delivery completion time τ_D and algorithm running time τ_{CPU} are considered. Maximizing the overall network performance and minimizing delivery completion time are two conflicting objectives for the problem under study. Thus, given the conflicting nature, it is essential to obtain the set of best trade-off or Pareto optimal solutions, in which an improvement in one objective is not possible with out degrading the other. In addition, when we consider large amount of delivery tasks or robotic vehicles, the running time of the mathematical models increases exponentially. Hence, for each truck, we only consider a maximum of 2 drones and 2 SADRs on executing the delivery task. We compare various *offline* state-of-the-art parcel delivery shortest path calculation algorithms such as [9], [20] and [25] to select most appropriate algorithm for calculating the optimal delivery time. As shown in Table I, by considering both τ_D and τ_{CPU}, we choose the JOCR-U algorithm to determine the route of the robotic vehicles as well as to provide the time constraints. Specifically, given the total operation time of all robotic vehicles T, τ is calculated given by:

$$\tau = T - \max(\sum_{n_d, n_s} (\tau_D(\sum_{u,v \in E} t_{uv}^{T})) + \hat{\tau}_{CPU}(\min \tau_{JOCR}^{U})) \tag{5}$$

where $\hat{\tau}_{CPU}$ is the normalized CPU time measured in seconds for running the JOCR-U algorithm. Also, n_d and n_s represents the number of tasks assigned to the drones and SADRs.

TABLE I
MAXIMUM DELIVERY COMPLETATION TIME AND CPU TIME ON COMPUTATION FOR JOCR-U, MFSTSP AND HGA MODELS.

	N=1		N=5		N=10	
	τ_D	τ_{CPU}	τ_D	τ_{CPU}	τ_D	τ_{CPU}
JOCR-U	**5.5**	**70**	**6.7**	**324**	**8.3**	8,117
mFSTSP	6.2	117	7.6	374	9.9	**7,703**
HGA	6.3	74	6.7	850	8.3	13,446

IV. VEHICLE NETWORK PATH COMPUTATION ALGORITHM

In this section, we detail three coupled algorithms that we propose in order to calculate the robotic vehicle paths for the delivery application problem: (i) the Offline Policy-based Algorithm (*ENCORE-offline*), (ii) the Online Heuristics-based Algorithm (*ENCORE-Heu*), and (iii) the Online Learning-based Algorithm ((*ENCORE-RL*)). Different strategies are used within each algorithm in order to improve the performance of the last-mile parcel delivery. Specifically, in *ENCORE-offline*, a 3D construction model is used to detect the blockage (i.e., obstacle features) between each two nodes. In contrast, the heuristics-based algorithm design uses a localized DDPG algorithm to improve the prediction accuracy of robotic vehicle trace paths; whereas, the learning-based algorithm design uses a Reinforcement Learning module to make near-optimal decisions on vehicle traces as well we the communication paths based on environmental awareness. Figure 2 shows a general workflow of the usage situation of these three algorithms inside one parcel delivery application.

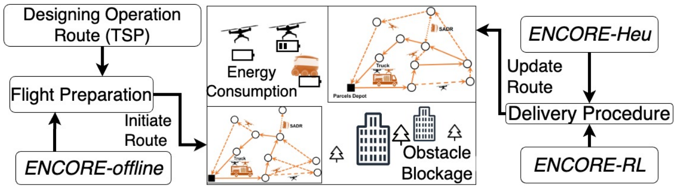


Fig. 2. Workflow to apply the three *ENCORE* algorithms during the flight preparation and delivery stages.

A. *Offline Policy-based Algorithm (**ENCORE-offline**)*

To address the time-efficiency versus continuous connection trade-offs in hybrid truck-drone-robot architectures, there is at first a need for a flexible policy-based node routing, i.e., a geographic routing protocol variant for robotic vehicles and a GCS. The location of the robotic vehicles should handle the frequently changing obstacle scene in urban scenarios within a fixed infrastructure by also adhering to time constraints. A choice of stateful and stateless routing protocol design is necessary to be determined based on the cost of maintaining routing tables versus the infinite loops in routing due to the local minima selection. In most of the high mobility power-constrained devices routing protocol design, the cost of maintaining node position in a database and updating the rapid changing status (intermittent failure of network, obstacle status) can cause robot-side time consumption drain. This can in turn lead to the failure of deliveries and even cause the robotic vehicles to crash. Consequently, in the policy-based algorithm i.e., *ENCORE-offline* design, we choose a stateless routing protocol in our base design, and on-demand communication is performed when requested. Moreover, the policy could influence the decision during the delivery task within a threshold. This is helpful in a case where a robotic vehicle finishes a single delivery task, and still has enough residual time/energy to safely return to the warehouse. To this end, there is a possibility that SADRs or drones may experience failure on delivery, while on the contrary, there is a higher chance for them to return back to the truck.

Policy is specified in the form of an time-oriented (small θ value), or communication/connection-oriented (large θ value) user preference. A warehouse manager may choose a time-oriented policy to avoid full network reconstruction when robotic vehicles face obstacles, in order to save time that could be used to achieve higher throughput of delivery tasks. Alternately, the warehouse manager can choose to monitor the vehicle traces to make sure the delivery task is on the track. However, based on the evaluation experiments in Section V, *ENCORE-offline* is shown to be able to continuously generate promising and stable communications while sacrificing only 10% of the on-board operation time of a robotic vehicle.

With the help of satellite images, it is relatively easy to learn the geo-information on existing physical obstacles before the delivery task is initiated. Thus, we leverage coordinates in space obtained via a GPS model embedded in moving nodes (i.e., drones, SADRs and trucks), to improve the precision of geographic routing. According to Equation 1, the policy-based algorithm provides a routing solution with flexible policy specifications to handle dynamic network situations. This in turn, determines various θ values, while addressing trade-offs in user preferences on either network throughput and residual operation time of robotic vehicles. The whole purpose of the *ENCORE-offline* algorithm is to find the shortest path from the sender s to the receiver r that could avoid the obstacle path on the route. At the beginning, each node only has knowledge of its surrounding nodes and the obstacle locations. In addition, policies are also provided at the beginning of the application to avoid further calculation when faced with limited time.

B. Online Heuristics-based Algorithm (ENCORE-Heu)

The performance of *ENCORE-offline* that involves average calculation time is around 7 seconds per request (see Table I). Consequently, by using this solution for rerouting information, the rescheduling of drone tasks may be delayed. This situation could influence the accuracy of autonomous drone control for last-mile parcel delivery. Thus, in order to achieve soft real-time cooperative drone management, dynamic models need to be used in online algorithms that improve the *ENCORE-offline* algorithm. In the online heuristics-based algorithm (*ENCORE-Heu*) design, we use an *off-policy* actor-critic network that follows DDPG [26] method to predict the relevant position between each two nodes in terms of A2A, A2G and G2G links, and heuristically determine the routing and overall time consumption in the delivery process.

Robotic Vehicle Trajectory Estimation: To estimate the trajectories of each robotic vehicle, we consider a simple hybrid drone-truck-robot environment with $D_{(n1)}$ drones, $D_{(n2)}$ SADRs and $T_{k(n)}$ trucks. $(n1, n2 \neq 0)$. We formulate the trajectory prediction problem as a partially observable Markov decision process (POMDP) [27] which is defined by the tuple containing the following -

$$M_{traj} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \mathcal{Z}) \tag{6}$$

where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, $\mathcal{R}$, $\mathcal{O}$ and $\mathcal{Z}$, are the state space, the action space, the probability of transition of states, the reward function, the observations and the probability distribution function for observed states, respectively. The POMDP aims to maximize the cumulative rewards that are received by the drones along their trajectories during the operation. The drones and SADRs are assumed to be fully charged before they leave truck for delivery. The learning environment comprises of one robotic vehicle with the truck following states $s_t = (P_{D_{(n1)}}, P_{T_{(n)}}, \phi_t, E_{t_n}, F_{t_{(n)}}, L_{d_{(n)}})$ representing robotic vehicle's position, truck's location, vehicle's heading, vehicle's total time capacity, truck's fuel capacity and delivery location, respectively and $s_t \in \mathcal{S}$ where $\mathcal{S}$ is the global state-space in the environment consisting of robotic vehicles and trucks. The truck is operated by a human driver, and in contrast, the robotic vehicles are autonomous and have to perform their delivery operations efficiently by following the set of actions defined as $a_t = (i)a_{(T)}$ - *'Take-Off/Leave from assigned truck T_k'*, $(ii)a_{(D)}$ - *'Drop-off Parcel'* and $(iii)a_L$ - *'Land-on/Return-to nearest truck T_k'*, $a_t \in A$, where A is the action space. Moreover, the rewards r_t are defined as follows: $+100$: successful parcel drop; $+10$: flying or moving towards $L_{d_{(n)}}$; -10: flying or moving away from $L_{d_{(n)}}$; -50: Losing parcel mid-mission; -100: collision with obstacles/drones;

Ensuing the design of the robotic vehicle's optimal trajectory selection and delivery completion scenario using an POMDP, the overall performance can be evaluated by tuning the values of the discount factor γ for establishing an optimal policy $\pi_t^* : S \rightarrow A$, which maps the states with best suitable actions. Let e_t be the episode during which the robotic vehicle performs a_t in the given S. The observations $o \in \mathcal{O}$ in any given episode e_t include the positions of other robotic vehicles and locations of other trucks that are observable to a given robotic vehicle and given truck, $o_t = \sum_i^U \sum_j^V \sum_j^W (P_{D_{n1(i)}}, P_{D_{n2(i)}}, P_{T_{(j)}})$, where U and V are total number of robotic vehicle and W is the total number of trucks, respectively and $o_t \sim \mathcal{Z}$. The drone or SADR receives r_t corresponding to the independent actions performed accordingly in given states as per time constraints of the fuel capacity of trucks. If the robotic vehicles successfully drop parcels, the actions reap higher rewards as they find the nearest trucks and return back after the delivery task. The POMDP aims to establish an optimal single-control policy π^* to maximize the value function V^π, given by:

$$V^\pi(s) = \mathbb{E}[\sum_{i=0}^{\infty}(\gamma^{i-t} r_i | s_t, \pi_t)]. \tag{7}$$

The proposed solution aims to solve the POMDP problem using an *off-policy* actor-critic network that uses the DDPG method such that the best possible actions are chosen in the state space of the hybrid drone-truck-robot environment. The action value function is given by:

$$Q_\pi(s_t, a_t) = \mathbb{E}[\sum_{n=0}^{\infty} \gamma^n r(s,a) | s = s_t, a = a_t] \tag{8}$$

The optimal policy, according to the Bellman's equation is given by -

$$\pi_t^* = \arg\max r_t(s_t, a_t) + \gamma \int_{s_{t+1}} V^\pi(s_{t+1}) ds_{t+1}$$

Heuristic Algorithm design: Based on the robotic vehicle trajectory estimation model, we can assume the local relative distance between the drones, between drones and SADRs, among SADRs, among drones and trucks in advance. Thus, to increase the accuracy of the robotic vehicle path computation procedure, our heuristics-based algorithm proactively performs a greedy calculation of the local-optimal path solutions.

Algorithm 1 shows the heuristic algorithm design utilizing the robotic vehicle trajectory estimation model detailed above. The main purpose of the algorithm is to utilize the prediction of the relative position of robotic vehicles and trucks. The time period t in the algorithm is default set to 10 seconds (line 20). For each time period, all θ values are calculated for each alternate neighbour node (line 26-29). Following this, we will calculate the performance gain on the drone (i.e., throughput gain by processed unit time) of each θ value (lines 2-17), and perform a greedy select of the highest performance gain (lines 23-25). Following this, we will use this θ value until the next time period t, when the heuristics-based algorithm is reinvoked (lines 18-20). Consequently, our heuristics-based algorithm will always select the local optimal choice, which however may not be the overall best of the entire system. Moreover, if the total required time is used up, the heuristics-based algorithm will shift to the *ENCORE-offline* with the corresponding time oriented θ value (lines 21-23). Hence, the heuristics-based algorithm performance may not provide impressive improvement from the offline *ENCORE-offline* solution in low obstacle scenarios. To this end, an online learning-based algorithm is needed to apply real-time decisions on the trajectories and communication barriers.

C. Online Learning-based Algorithm *(ENCORE-RL)*

Since *ENCORE-Heu* provides local optimal choice by time period, it may lack information on the global optimal performance gain over the entire system. Herein, we abstract the whole real-time robotic vehicle parcel delivery problem as a Markov-decision process (MDP), which can provide a mechanism to judge the θ choices by using rewards. Thus, the optimization problem can be redefined as: *find the optimal moving trace which minimizes the global performance gain for the robotic vehicle path computation process*. This finite-time MDP problem can be proposed as follows:

$$M_{env} = (S_{env}, A_{env}, P_{env}, R_{env}, M_{traj}) \tag{9}$$

where, S_{env} is the state space, A_{env} is the action space, P_{env} is the probability function that indicates the probability of action a in state s at time t will lead to state s' at time t+1. R_{env} is a reward function, and M_{Traj} is the optimal MDP result given by the *robotic vehicle trajectory estimation* algorithm.

Algorithm 1: ENCORE-Heu

```
Input: $[sensorSets]$:= multi-sensor dataset from all given neighbor nodes;
       $T$:= time budget; $D_{t(n)}$:= number of drones; $T_{k(n)}$:= number of
       trucks; $Q(s_t|\theta^\mu)$:=Actor network; $Q(s_t, a_t|\theta^c)$:= Critic network;$\theta^\mu$
       $\theta^c$:=weights; $(s_t, a_t, r_t, s_{t+1})$:= replay buffer B
Output: $N_x$:= the exact neighbour to send packets for the next $t^{th}$ time;
        $[policyGains]$:= time and throughput gain from the connection;
        Optimal Policy:= given accuracy on prediction based on Actor-Critic
        network
1  Function Main():
2      for $e_t = 1, 2, ...N$ do
3          while not done do
4              Perform $a_t$ in given $s_t$ and receive $r_t$;
5              store transitions $(s_t, a_t, r_t, s_{t+1})$;
6              update $Q(s_t, a_t)$
7              obtain policy $\pi_t$ in actor network;
8              obtain value function $V^\pi$
9              Calculate the Loss for actor:
                 $L(\theta^\mu) = \frac{1}{N}\sum_{i=1}^{n}[-Q(S(i), \pi_{s_t}(i); \theta^\mu)]^2$;
10             Calculate the Loss for critic:
                 $y_t(i) = r_t + \gamma \cdot Q^{'}(s_{t+1}, \pi_t^{'}(s_{t+1}; \theta^{'\mu})|\theta^{'c})$;
11             $L(\theta^c) = \frac{1}{N}\sum_{i=1}^{n}[y_t(i) - Q(s_t(i), a_t(i); \theta^c)]^2$
12             update policy $\pi_t^*$ in actor network;
13             Update $V^\pi(s)$;
14             optimal policy $\pi_t^*$;
15         sample random training examples from B;
16         Gradient updates for the actor critic target networks:
17         $\theta'^\mu \leftarrow \theta^\mu$, $\theta'^c \leftarrow \theta^c$
18     if $T$ then
19         $result \leftarrow$ HEURISTIC($[newNodeSet], newS$)
20         $T \leftarrow T - t$
21     else
22         $result \leftarrow$
23         ENCORE-OFFLINE($[newNodeSet], [newObsSet], newS, \theta$)
24     $send(P, N_{result})$
25     $Calculate[policyGains]$
26 Function Greedy($[newNodeSet], newS$):
27     foreach $\theta$ do
28         $N_a \in argminf(n, P_{newS.x,y,z}, \theta)$
29     return $\max N_a$
```

To ensure that the near optimal θ can be chosen at every step, we consider the dynamic decision-making problem to be used for downstream tasks at time t. The situation will turn to an time-oriented case, when action goes to -1. At the same time, the situation will turn to a throughput-oriented case, when action goes to 1 in terms of the step size δ value chosen. Given that when δ value is small, the system will take more resources (i.e., energy and time) to calculate the optimal value, and each action change may only produce small reward gains. Thus, we require the δ value selected to be less than or equal to 0.5, and greater than 0.15 to avoid the void and eliminate redundant calculation. Formally, the actions of the MDP problem can be discretely given by the equation:

$$a^t_{env} = \begin{cases} -1_{time\_orient}, \hat{f} = f(\theta - \delta), \theta \geq \delta \\ 0_{keep\_the\_same\_state}, \hat{f} = f(\theta) \\ 1_{throughput\_orient} \hat{f} = f(\theta + \delta), \theta \leq 1 - \delta \end{cases} \tag{10}$$

Specifically, we define the state of the MDP problem to be chosen between actions with a given time period. The state contains the stored previous predictions f results, and the remaining query budget on both parcel-delivery-time-oriented as well as throughput-oriented. The state in the MDP can be formalized as shown in the following equation:

$$s^t_{env} = [\theta_t, f_{previous}, f_{(\theta-\delta, \theta+\delta)}, M_{traj}] \tag{11}$$

The reward function is given to minimize the cost of a single robotic vehicle operation time τ and obstacle-awareness recovery time δ given in Equation 1. Consequently, we considered the reward function to be the same as in Equation 1, with respect to operation time consumption on a single robotic vehicle and theoretical guarantees on packet delivery in obstacle situations. Thus, we can ultimately define the reward function as follows:

$$R^t_{env}(s^t, a^t) = -\alpha \cdot \underbrace{\tau(f^t, \hat{f}^t)}_{residual\,time} - \beta \cdot \underbrace{\delta(cost(a^t))}_{communication} \quad (12)$$

Having defined the environmentally aware parcel delivery scenario as an MDP, we can evaluate the overall performance by minimizing the expected total reward the system achieves. In other words, we can state: Given the choice of 5 discrete levels of routing model f, a optimal trajectory prediction overhead M_{traj}, and a MDP problem M_{env}, find optimal routing policy $\pi_{env} : S_{env} \rightarrow A_{env}$ that maximizes expected cumulative reward R_{env}:

$$\pi_{env} \in argmax\mathbb{E}(\sum_{T} R_{env}(s_{env}, a_{env})) \quad (13)$$

Although we framed this problem as an MDP, it is not easy to apply conventional techniques such as dynamic programming for solving the problem. This is because, many of the aspects of this problem are hard to analytically characterize, especially the dynamics of the sensory input stream (e.g., GPS, obstacle, dynamic flight time, dynamic SADR operation time). This motivates our integration of a soft optimal solution that uses a model-free Reinforcement Learning technique. The reason to use such a technique is as follows: it is capable of learning optimal discrete policies based solely on the features included in the state, and avoids the need to predict the future states (as done in our heuristics-based algorithm). Specifically, we use the state-of-the-art multi-agent Q-learning algorithm [28], which is easy to deploy, efficient to evaluate in terms of dynamics, and is amenable to effectively perform optimization-based action selection.

V. Performance Evaluation

In this section, we first introduce the evaluation setup and data sets collected from a hybrid drone-truck-robot last-mile parcel delivery use case scenario [29]. Next, we discuss the policy-based, heuristics-based and learning-based results on cooperative vehicle mobility management. The results are compared with Non-AI aided, AI-aided and Oracle solution, and a comprehensive solution by considering together the accuracy and time metrics. Lastly, we present our findings and use cases which could leverage various solutions and the obtained simulation results for those approaches.

A. Experiment setup

For evaluation of our parcel delivery framework, we initialized a simulated urban environment using 3D building models in ns-3 simulator based on an urban road map. The simulation script randomly generates locations of warehouse, customer, and corresponding delivery capacity on parcels in the range of an urban map. The simulator scripts can change the density of the buildings to simulate various urban maps from a small city center to a metropolitan downtown. Specifically, users can assign: (i) number of trucks used for intermediate communication and transportation, and (ii) number of robotic vehicles (with a selection on the amount of drones and SADRs), one truck could carry-on (based on our setting, one truck can only carry at most 2 drones and at most 3 SADRs). Following this, the simulator scripts calculate the initial route for each drone based on the JOCR-U algorithm. Table II shows the basic setting of the simulations and the altitude of drone in consideration varies from 0 to 50 m. In the setting of large buildings, the size is closer to the connection range between the drones. Width of an obstacle (e.g., building) is set to be between 50 to 100 m. Parcel weight is restrict to 5kg for drones, and 10kg for SADRs. To run the experiments in a reproducible and reliable testbed, we use a trace-based drone-robot-edge simulation platform that we developed on top of ns-3. This platform integrates simulation on among drone, SADR, truck and networking sides, and provides flexibility in adding plugins for drones, SADRs and GCS to e.g., change the mobility model on the robotic vehicles, adding multi-sensor simulations and applying realistic map interfaces. For each setting, we run the experiments in 5 different low density architecture scenarios (i.e., 10% of the obstacle exists which in turn causes 20% of blockage of A2G and G2G links in the whole map) and 5 high density architecture scenarios (i.e., 10% to 60% obstacle density). In addition, this platform provides traces generated in every experiment, which was used in our Reinforcement Learning training procedures.

TABLE II
Environment Settings for Simulation

Application Settings		**Network Settings**	
No. of vehicle:	1-3	Transport protocol:	RUDP
Delivery area:	10-15 miles	Application Bit rate:	6 Mbps
Obstacle size:	60*30*20 m	Tx power:	32-48 dBm
Radio range:	250 m	Tx/Rx gain:	3 dB
Drone dist.:	Euclidean	Prop. model:	TWO RAY
Truck/SADR dist.:	Rectilinear	Max. msg. size:	4000 bit
Simulation time:	7000-8000 s	WIFI protocol	802.11s
Avg. vehicle speed:	10 - 35 mph	Modulation:	OFDM
Parcel weight:	5kg, 10kg	Data rate:	65 Mbps

Composition Metrics. For the communication measurement, we measure the average *throughput* level in Mbps of the packet transmission channel on all the A2G, A2A, and G2G links. Higher the throughput given, higher is the network performance the algorithm can achieve. In addition, to test the GCS's ability to use *ENCORE-offline* and adapt to policy changes, we evaluate our Reinforcement Learning module on each trace for different δ values and for every query budget fraction in $\delta \in [0.15, 0.25, 0.5]$, simulated on different reward function weights.

For the time measurement, we use a simple time consumption model, in which every drone starts with the same flight time budget (i.e., 1000 seconds), and every SADR starts with the same operation time budget (i.e., 2500 seconds). The consumption is calculated during the whole parcel delivery procedure. Next, we calculate the *residual time* by averaging the individual residual vehicle operation time of all the robotic vehicles in one experiment. We remark that the time used for transmission is comprised of the times used for the algorithm recalculation and network establishment. However, the realistic operation time could exceed the given time which may influence the *success ratio* of a given task. For example, if a drone takes a long time for receiving the updated trace

TABLE III

ENCORE-offline PERFORMANCE COMPARISON ON THROUGHPUT (MBPS)

	Drone	M-GEAR (with θ = 0,1)		*ENCORE-offline* (with θ = 0,0,25,0,5,0,75,1)					O-HWMP
low	1	$0.89{\pm}0.39$	2.48 ± 1.71	2.07 ± 0.77	1.93 ± 0.62	2.18 ± 0.60	2.84 ± 0.81	**3.17±1.18**	0.83 ± 0.22
	3	$1.70{\pm}0.72$	2.19 ± 0.98	1.95 ± 0.57	2.15 ± 0.57	2.25 ± 0.57	2.64 ± 0.51	**2.59±0.52**	0.76 ± 0.22
high	1	$0.85{\pm}0.45$	2.36 ± 1.84	1.56 ± 0.95	1.62 ± 0.88	2.05 ± 1.07	2.87 ± 1.36	**3.01±1.57**	0.77 ± 0.31
	3	$1.65{\pm}0.71$	1.95 ± 0.82	1.94 ± 0.55	2.04 ± 0.56	2.31 ± 0.56	**2.40±0.73**	2.31 ± 0.37	0.71 ± 0.23

TABLE IV

ENCORE-offline PERFORMANCE COMPARISON ON DELIVERY COMPLETION TIME (S)

	Drone	M-GEAR (with θ = 0,1)		*ENCORE-offline* (with θ = 0,0,25,0,5,0,75,1)					O-HWMP
low	1	10.25±0.7	10.22±0.46	10.19±0.34	10.20±0.37	10.19±0.42	10.22±0.37	10.20±0.31	**9.83±0.61**
	3	8.14±0.14	8.15±0.13	8.14±0.12	8.07±0.18	8.06±0.18	8.06±0.50	8.01±0.40	**7.32±0.89**
high	1	12.02±0.52	11.97±0.47	12.03±0.49	11.98±0.36	11.9±0.40	**10.00±0.39**	10.10±0.40	10.00±0.50
	3	10.10±0.13	9.99±0.16	10.04±0.48	10.04±0.46	9.95±0.40	**9.93±0.39**	**9.93±0.39**	10.01±0.86

information packets from the ground nodes, it will waste time hovering or will take a deflected flight course, which may ultimately result in a delivery task failure.

B. Oracle and Baseline Solution Approaches

To design a better routing approach for the hybrid drone-truck-robot system, we compare the delivery efficiency and the throughput performance with existing state-of-the-art algorithms and also the Oracle result. All experiments are using the same dataset which is detailed earlier in Section V-A.

Non-AI-aided Stateless Protocol Baselines π_{path}^{M-GEAR} and π_{path}^{O-HWMP}. Both baselines provide stateless protocol approaches which implement ad-hoc network path computation in either air and/or ground links between mobile devices and ground nodes. We compare our *ENCORE-offline* approach with both non-AI protocols, viz., Gateway Based Energy-efficient Routing Protocol (M-GEAR) [30] and the Optimized-Hybrid Wireless Mesh Protocol (O-HWMP) protocol 802.11s standard [31]. Since each protocol follows different aspects of path computation strategies (e.g., O-HWMP, as a updated version of RM-AODV [32] do not consider time constraints), it is hard to compare side-by-side performance with each of both approaches. However, M-GEAR as a variant of Geographical and Energy Aware Routing (GEAR) [33], is the most appropriate baseline for comparison for two reasons: (i) it considers resources energy constraints with given geo-spatial information, which the energy consumed can be converted into the time spent on drone in proportion, and (ii) it provides simple geo-location based routing procedures.

AI-aided Stateless Protocol Baselines π_{path}^{PARRoT}. This baseline utilizes AI techniques to enhance the performance of the routing protocol design. Compared to the *ENCORE-Heu* and *ENCORE-RL* approach we proposed, this predictive ad-hoc routing fueled by reinforcement learning and trajectory knowledge (PARRoT) [34] also considers obstacle blockage on communication, and uses the Q-learning algorithm to predict the trajectory of the drones. Both urban and rural scenes are considered in PARRoT. However, based on the settings relevant to our proposed application, we only compare with the urban scene that the PARRoT estimated.

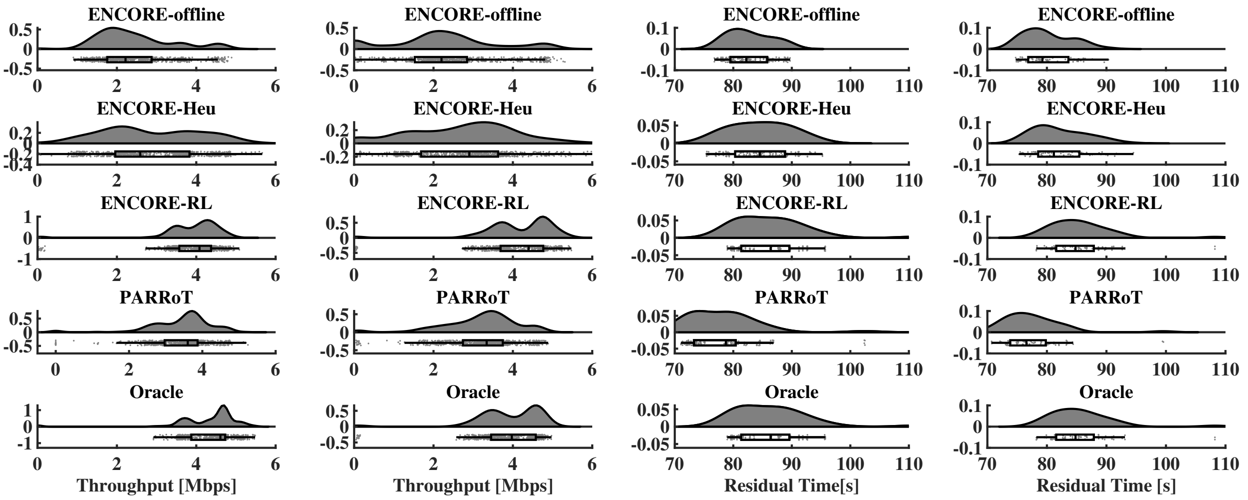
Oracle approach π_{path}^{Oracle}. The Oracle baseline is given by the optimized robotic vehicle operation path in advance by considering: (i) the physical obstacle information among A2A, A2G and G2G communication links relative to a drone or a SADR, and (ii) the exact vehicle position on a time-by-time basis with relevance to all GCS nodes (e.g., trucks). This approach cannot be achieved when the environment is unknown or if a new delivery position is added.

C. Evaluation results

Considering only offline approaches, *ENCORE-offline* and O-HWMP are Pareto-optimal path computation strategies. For every θ value given in Equation 1, Table III shows that *ENCORE-offline* outperforms related GEAR and AODV protocols in terms of the throughput level metric. Further, Table IV shows that *ENCORE-offline* achieves comparable or even better results in terms of time consumption. However, O-HWMP shows better delivery time efficiency than *ENCORE-offline* due to its ability to use spanning trees for minimizing the number of control messages on the one robotic vehicle scenario. Note that spanning tree approaches perform worst compared with other protocols under high failure scenarios, which are common in multiple robotic vehicle scenes. Thus, we can conclude that both *ENCORE-offline* and O-HWMP are Pareto-optimal robotic vehicle path computation strategies in hybrid drone-truck-robot parcel delivery, if no online information query and learning procedures exist. In other words, both solutions have no alternative solutions that make any one preference criterion better-off without making at least one preference criterion worse-off.

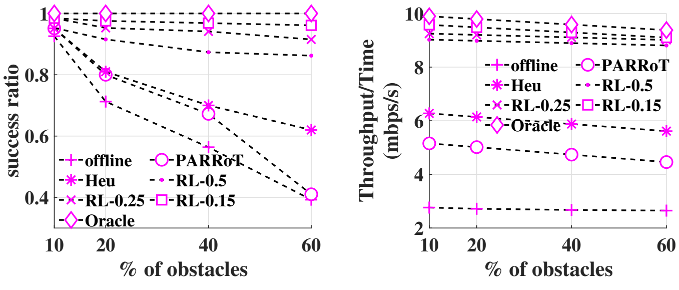
***ENCORE-offline* performs suitably in offline only approaches.** Due to local geo-spatial knowledge of the delivery route, *ENCORE-offline* shows the most promising performance compared with other geographic routing approaches (e.g., M-GEAR) as shown in the throughput standard deviation values in Table III. Even through O-HWMP has advantages over proactive stateful routing solutions, they do not show advantages in highly dense mobility delivery scenarios (i.e., with high operation speed and limited delivery time) in terms of acceptable throughput levels.

ENCORE-offline results from Tables III and IV can be summarized as follows. We can see two major observations in the single robotic vehicle (one drone and one SADR), single truck scenario, when comparing low network failure (10% of blockage obstacles in a map) and high network failure (60% of blockage obstacles in a map) cases. In the first observation, throughput as well as delivery completion times are higher in low network failure scenes. This is because - it takes time to calculate the routing path on each robotic vehicle, and the vehicles will wait for transmission rebuild events. Moreover, in the high network failure scenario, this operation happens more frequently than in the lower network failure scenario. In the second observation, on both high and low network



(a) Throughput on low obstacle scenes. (b) Throughput on high obstacle scenes. (c) Residual time on low obstacle scenes. (d) Residual time on high obstacle scenes.

Fig. 3. Horizontal comparison of three approaches with PARRoT and oracle solution in terms of throughput distribution and residual time on each drone.



(a) *Success ratio*

(b) *Performance gain*

Fig. 4. *Success ratio* and *Performance gain* (throughput gain by processed time) of various approaches. Parameter with *RL* represents different δ value choice in reward function.

failure scenes, single vehicle throughput performance varies significantly considering different θ values. This is because - in a single robotic vehicle scenario, the recovery time will be longer than in the case of the multiple robotic vehicle scenario in terms of the value of $\tau(n, d)$ going high, and the performance on average throughput being relatively low when $\tau(n, d)$ is considered.

***ENCORE-Heu* solution could be an alternative choice in high density urban area scenarios w.r.t. connection and resources.** From the results in Figures 3a and 3b, we can see that the *ENCORE-Heu* solution performance is different on low network failure cases (with 10% of obstacles in a map) and on high network failure cases (with 60% of obstacles in a map). This is due to the fact that high network failure cases normally consist of sequential architectures, where the *ENCORE-Heu* solution could provide guidance for the future location of robotic vehicles. On the other hand, a wholesome infrastructure setup in metropolitan areas could provide considerable sensor resources that could highly increase the accuracy of our robotic vehicle position model used in the *ENCORE-Heu* approach. Although *PARRoT* provides higher network performance under similar learning-based trajectory prediction algorithm, *ENCORE-Heu* could achieve significant time saving than the *PARRoT*, which can be seen in both Figures 3c and 4b. Moreover, compared with the *ENCORE-RL* approach, the *ENCORE-Heu* solution does not require significant computation resources. For example, we used AWS SageMaker [35] to train the learning model and generate policies. Once the tasks or the percentage of obstacles occupation increases, the cost of time and budget is exponentially increased. Thus, if there are budget limitations or if we are addressing repetitive delivery tasks, the *ENCORE-Heu* solution is a better choice.

Our approach - *ENCORE* outperforms in terms of effectiveness and efficiency while solving the robotic vehicle parcel delivery problem. By observing Figures 3 and 4 together, we can make the following interesting conclusions. First, if we are not concerned about the lack of robotic vehicles in an urban area, or the lack of computation resources in a warehouse, *ENCORE-offline* and O-HWMP are two off-line Pareto-optimal choices for warehouse managers. On the contrary, when we are concerned with more effective approaches (that increase the number of deliveries) in a metropolitan area, online approaches such as the *ENCORE-Heu* algorithm can provide a time-saving choice in terms of learning model training. Finally, when a cost requirement is not important, the *ENCORE-RL* approach with low δ values is most suited as seen in Figure 4. Moreover, *ENCORE-RL* achieves between 85-96% of the Oracle strategy performance. We remark that a lower δ value achieves more effectiveness (as seen in the higher delivery success ratios in Figure 4a) and higher efficiency (as seen in the high performance gains in Figure 4b).

VI. Conclusion

In this paper, we proposed and evaluated novel, coupled robotic vehicle communication path computation algorithms that use environmental awareness for the last-mile parcel delivery problem. Using trace-based simulations on multi-drone, multi-truck optimal pre-application schedulers, we have shown that our *ENCORE-offline* approach Pareto-optimally computes efficient paths w.r.t. time and communication trade-offs, with out being aware of the environment in advance. Moreover, comparisons of proactive and reactive protocols showed that our online *ENCORE-RL* algorithm achieves between 85-96% of performance compared to the Oracle solution on average. In addition, our approach is flexible to be implemented across a range of small suburbs to large metropolitan areas with different number of obstacles, and has a performance gain of 98% in terms of throughput versus time, in comparison with the Oracle strategy.

References

[1] R. Events, "Last mile delivery in america." [Online]. Available: https://tinyurl.com/333xmtfy

[2] W. E. Forum, "The future of the last-mile ecosystem." [Online]. Available: https://tinyurl.com/44pmefa6

[3] J.-P. Aurambout, K. Gkoumas, and B. Ciuffo, "Last mile delivery by drones: An estimation of viable market potential and access to citizens across european cities," *European Transport Research Review*, vol. 11, no. 1, p. 30, 2019.

[4] M. Salama and S. Srinivas, "Joint optimization of customer location clustering and drone-based routing for last-mile deliveries," *Transportation Research Part C: Emerging Technologies*, vol. 114, p. 620–642, May 2020.

[5] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transportation Research Part C: Emerging Technologies*, vol. 110, p. 368–398, Jan 2020.

[6] D. Jennings and M. Figliozzi, "Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 6, p. 317–326, May 2019.

[7] M. A. Figliozzi, "Carbon emissions reductions in last mile and grocery deliveries utilizing air and ground autonomous vehicles," *Transportation Research Part D: Transport and Environment*, vol. 85, p. 102443, Aug 2020.

[8] C. W. Chen, "Drones as internet of video things front-end sensors: challenges and opportunities," *Discover Internet of Things*, vol. 1, no. 1, pp. 1–12, 2021.

[9] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 368–398, 2020.

[10] S. Abdolinezhad, M. Schappacher, and A. Sikora, "Secure wireless architecture for communications in a parcel delivery system," in *2020 IEEE IDAACS-SWS*. IEEE, 2020, pp. 1–6.

[11] M. Y. Arafat and S. Moh, "Location-aided delay tolerant routing protocol in uav networks for post-disaster operation," *IEEE Access*, vol. 6, pp. 59 891–59 906, 2018.

[12] A. Khochare, Y. Simmhan, F. B. Sorbelli, and S. K. Das, "Heuristic algorithms for co-scheduling of edge analytics and routes for uav fleet missions," *arXiv preprint arXiv:2102.08768*, 2021.

[13] 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*, pp. 3098–3105, 2020.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[15] 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.

[16] T. Nguyen and T.-C. Au, "Extending the range of delivery drones by exploratory learning of energy models," in *Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2017, pp. 1658–1660.

[17] J. Vepsäläinen, "Energy demand analysis and powertrain design of a high-speed delivery robot using synthetic driving cycles," *Energies*, vol. 15, no. 6, p. 2198, 2022.

[18] D. Baek, Y. Chen, A. Bocca, A. Macii, E. Macii, and M. Poncino, "Battery-aware energy model of drone delivery tasks," in *Proceedings of the International Symposium on Low Power Electronics and Design*.

[19] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *International Journal of Production Economics*, vol. 225, p. 107598, 2020.

[20] M. Salama and S. Srinivas, "Joint optimization of customer location clustering and drone-based routing for last-mile deliveries," *Transportation Research Part C: Emerging Technologies*, 2020.

[21] C. Qu, R. Singh, A. E. Morel, F. B. Sorbelli, P. Calyam, and S. K. Das, "Obstacle-aware and energy-efficient multi-drone coordination and networking for disaster response," in *2021 IEEE CNSM*, pp. 446–454.

[22] A. Rovira-Sugranes, A. Razi, F. Afghah, and J. Chakareski, "A review of ai-enabled routing protocols for uav networks: Trends, challenges, and future outlook," *Ad Hoc Networks*, vol. 130, p. 102790, 2022.

[23] K. Mandal, S. Halder, P. Roy, M. K. Paul, S. D. Bit, and R. Banerjee, "An online mobility management system to automatically avoid road blockage and covid-19 hotspots," *New Generation Computing*.

[24] A. Ferlini, W. Wang, and G. Pau, "Corner-3d: A rf simulator for uav mobility in smart cities," in *ACM SIGCOMM 2019 workshop*, ser. MAGESys'19, 2019, p. 22–28.

[25] K. Peng, J. Du, F. Lu, Q. Sun, Y. Dong, P. Zhou, and M. Hu, "A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery," *IEEE Access*, vol. 7, pp. 49 191–49 200, 2019.

[26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. PMLR, 2014, pp. 387–395.

[27] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[28] D. Tamagawa, E. Taniguchi, and T. Yamada, "Evaluating city logistics measures using a multi-agent model," *Procedia - Social and Behavioral Sciences*, 2010, the Sixth International Conference on City Logistics.

[29] C. Murray, "mfstsp source code dataset." [Online]. Available: https://github.com/optimatorlab/mFSTSP/tree/master/Problems

[30] Q. Nadeem, M. Rasheed, N. Javaid, Z. Khan, Y. Maqsood, and A. Din, "M-gear: Gateway-based energy-aware multi-hop routing protocol for wsns," in *2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications*, 2013.

[31] C. J. Katila, A. Di Gianni, C. Buratti, and R. Verdone, "Routing protocols for video surveillance drones in ieee 802.11 s wireless mesh networks," in *2017 European Conference on Networks and Communications (EuCNC)*. IEEE, 2017, pp. 1–5.

[32] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 1999, pp. 90–100.

[33] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," 2001.

[34] B. Sliwa, C. Schüler, M. Patchou, and C. Wietfeld, "PARRoT: Predictive ad-hoc routing fueled by reinforcement learning and trajectory knowledge," in *2021 IEEE 93rd Vehicular Technology Conference (VTC-Spring)*, Helsinki, Finland, Apr 2021.

[35] D. Hudgeon and R. Nichol, "Machine learning for business: using amazon sagemaker and jupyter," 2020. [Online]. Available: https://docs.aws.amazon.com/sagemaker/latest/APIReference