A reinforcement learning framework for improving parking decisions in last-mile delivery

Muriel, Juan E.; Zhang, Lele; Fransoo, Jan C.; Villegas, Juan G.

Published in:
Transportmetrica B: Transport Dynamics

DOI:
10.1080/21680566.2024.2337216

Publication date:
2024

Document Version
Peer reviewed version

A Reinforcement Learning Framework for Improving Parking Decisions in Last-Mile Delivery

Juan E. Muriel[ab1], Lele Zhang[cd], Jan C. Fransoo[e] and Juan G. Villegas[f]

[a]Pacific National, Brisbane, Queensland 4006, Australia

[b]School of Engineering and the Built Environment (SEBE), Deakin University, Geelong, Victoria 3216, Australia

[c]School of Mathematics and Statistics, The University of Melbourne, Parkville, Victoria 3010, Australia

[d]ARC Training Centre of Optimisation Technologies, Integrated Methodologies, and Applications

[e]Tilburg School of Economics and Management, Tilburg University, 5000 LE Tilburg, Netherlands

[f]Departamento de Ingeniería Industrial, Facultad de Ingeniería, Universidad de Antioquia, Calle 70 No. 52-21, Medellín, Colombia

Abstract:

The convergence of economic, social, and technological factors underscores the pressing need for urban last-mile delivery to be both swift and dependable while minimising adverse impacts on mobility, safety, and the environment. However, the limited availability of parking space for freight vehicles in large cities fails to meet the growing demand for urban freight services, emphasising the imperative for efficient and cost-effective solutions.

In this study, we leverage simulation-optimisation techniques in conjunction with a Reinforcement Learning (RL) model to analyse the routing behaviour of delivery vehicles (DVs) and provide them with the ability to adapt and learn from their delivery environment (i.e., road network). We conceptualise the system as a stochastic k-armed bandit problem, representing a sequential interaction between a learner (the DV) and its surrounding environment.

Before entering the system, each DV is assigned a random number of customers and devises an initial delivery route. Should it encounter an unavailable loading zone, it employs the RL model to select a delivery strategy, thereby modifying its route accordingly. The reward, or in our case, penalty, is gauged by the additional trucking and walking time incurred compared to the originally planned itinerary.

Our methodology is tested on a simulated network featuring realistic traffic conditions and a fleet of DVs employing four distinct last-mile delivery strategies. The results of our numerical experiments underscore the advantages of providing DVs with an RL-based decision support system for en-route decision-making, yielding benefits not only to courier companies but also to the overall efficiency of the transport network.

Keywords: last-mile delivery, urban logistics, reinforcement learning, loading zone, simulation-optimisation

Highlights:

- Combining simulation and optimisation algorithms with reinforcement learning
- Model DVs en-route parking decisions with a k-armed bandit algorithm
- Evaluating the impacts of delivery strategies on traffic congestion and in last-mile delivery efficiency

[1] Corresponding author. E-mail address: juan_muriel@pacificnational.com.au

1. Introduction

The demand for last-mile delivery is experiencing rapid growth on a global scale. According to Business Research Insights (2023), the last mile delivery market is projected to expand annually by nearly 10%, with developing economies anticipating a surge of over 300% (McKinsey & Co., 2019). This surge can be attributed to several key factors, including prevailing production and distribution practices and the significant rise in business-to-consumer deliveries (Crainic, Ricciardi & Storchi, 2004; Gevaers, Van de Voorde & Vanelslander, 2011). This trend is further aggravated by the escalating demand for international products, the decreasing lifespan of goods, and the constrained capacity coupled with escalating prices of warehouse sales floor space (Ewedairo, Chhetri & Jie, 2018). Even though last-mile delivery services have been available for a while, consumers now increasingly expect immediate delivery, often prioritising speed or convenience over operational efficiency (Casey et al., 2014).

To address these challenges, both practitioners and academics have proposed a spectrum of solutions spanning infrastructure management, vehicle-related strategies, traffic management, financial approaches, logistical management, demand and land use management, and parking management, including Loading Zones (LZs) (Holguin Veras et al., 2020). Among these, parking management, despite its apparent simplicity in terms of infrastructure, has emerged as a crucial strategy for effectively managing freight traffic (Shiftan & Burd-eden, 2001).

In the United States alone, the cost attributed to time lost due to parking-related congestion caused by freight vehicles approaches a staggering $10 billion, a figure rivalling only with vehicular crashes and adverse weather (Han et al., 2005). At the individual vehicle level, the seminal work of Shoup (2005) drew significant attention to cruising for parking as a major contributor to urban traffic congestion. His pioneering study, based on sample estimates, revealed that cruising for parking accounted for 30% or more of vehicle activity. Similarly, Barter (2013) found that, on average, 28% of intercepted motorists in New York City were engaged in the search for parking. While more recent studies utilising GPS trackers report figures of less than 5-6% (Weinberger, Millard-Ball & Hampshire, 2020), the problem could be exacerbated when considering delivery vehicles (DVs) cruising, the sudden surge in traffic volume and the limited space and configuration of LZs in central business districts (CBDs).

From the city's perspective, the long-term repercussions of DVs making suboptimal parking decisions contribute to a growing number of transport-related externalities, with unsustainable effects on people, the economy, and the environment. Presently, last-mile delivery (LMD) accounts for between 16% and 50% of air pollutant emissions (SO2, NOx, and CO) (Behrends, Lindholm & Woxenius, 2008), and according to the International Transport Forum (2017), global CO2 emissions from transport are projected to rise by 60% by 2050 despite significant technological advancements.

Despite substantial investments by courier companies to plan and optimise their operations, DVs encounter a myriad of challenges in locating available LZ. The dynamic nature of traffic conditions, fierce competition for LZs, and their limited availability not only represent logistical problems but also pose a significant urban planning challenge (Marcia, 2009). Incorporating crucial factors such as drivers' tacit knowledge—comprising familiarity with local geography, infrastructure, parking preferences, and consumer flexibility—into modern delivery management systems remains a daunting task. This creates a notable disparity between theoretical route planning and real-world execution, a gap that most optimisation-based approaches struggle to bridge (Amazon Last Mile Routing Research Challenge, 2021).

Recent studies highlight the considerable time DVs spend—averaging between 6 to 24 minutes—searching for an LZ (Dalla Chiara & Goodchild, 2020), often leaving parking decisions to the driver's discretion in the absence of decision-support systems (Boysen, Fedtke & Schwerdfeger, 2021). For instance, research conducted in Melbourne revealed that 53% of surveyed freight carriers rely on drivers to organise parcel deliveries based on their local expertise, with only 31% utilising routing and scheduling software (Aljohani and Thompson, 2018).

At the operational level, unsupported parking decisions during LMD lead to several detrimental outcomes. DVs may end up parking farther from commercial establishments, increasing human fatigue and extending route time and distance. Alternatively, they might circle aimlessly in hopes of securing a parking spot, exacerbating congestion and pollution. Moreover, resorting to double parking violates traffic regulations, heightening crew safety risks, worsening congestion, and impacting on visual amenity. These practices result in substantial expenses for courier companies, as evidenced by parking fines and towing fees. In 2019 alone, FedEx and UPS paid a staggering $9.8 million and $23 million, respectively, in fines for over half a million violations in New York City (Baker, 2019). Similarly, a study examining 374 vehicles found that 25% were unlawfully parked, with violations ranging from unpaid meters to non-compliance with parking signage and double parking (Jaller, Holguín-Veras & Hodge, 2013).

To address these challenges, the development and implementation of cost-effective LMD solutions are imperative. These solutions must empower drivers to make real-time parking decisions amidst the dynamic and nature of their environment, while also navigating the competition and interactions within the different actors of the road network (Dablanc, 2011; Bektas, Crainic & Van Woensel, 2017; Campagna et al., 2017; Zhang & Thompson, 2019).

In this paper, we delve into a road transport system where a group of learning agents, representing delivery vehicles (DVs), need to make en-route decisions when confronted with unavailable Loading Zones (LZs). We conceptualise this system as a stochastic k-armed bandit problem, akin to a sequential game between a learner (i.e., a DV) and its environment (i.e., the road network). Each iteration simulates the road network's dynamics over a typical morning period. In episodes throughout this period, should an LZ be unavailable, an agent must determine its subsequent course of action for efficient delivery.

This study explores various delivery strategies employed by DVs when faced with such decisions:

- *Alternative LZ* - finding the shortest path to the closest LZ,
- *Illegal Parking* - attempting to park illegally at the rear or front of the LZ,
- *Last Delivery* - leaving the delivery for the end of the route, and
- *Mixture of the above three* - we adopt reinforcement learning (RL) to assist the DVs in choosing the best strategy

The methodology employed in this paper extends upon the hybrid simulation-optimisation model introduced by Muriel et al. (2022) embedded within a RL framework. This approach combines stochastic cellular automata (CA) for simulating traffic patterns with a metaheuristic and a commercial solver to simulate the behaviour of both private vehicles (PVs) and delivery vehicles (DVs). It accounts for the decision-making processes of PVs and DVs, their interactions, and the inherent variability of stochastic parameters such as traffic conditions, competition, cruising, and illegal parking. Within the RL framework, each DV functions as an agent that learns from and interacts with the road network environment constructed by the simulation-optimisation model. Based on past performance, the DVs take actions (i.e., select strategies) aimed at optimising their delivery process. The reward signal in the RL framework, in this case, the penalty, is computed as the additional trucking and walking time relative to the optimal delivery route under ideal conditions, assuming no traffic congestion or competition.

The methodology is applied to a simulated network featuring realistic conditions, where the four last-mile logistics strategies discussed earlier are evaluated. The performance of these strategies is assessed based on both the courier's economic objective, specifically delivery efficiency in terms of total travel time, and the network's performance objective, focusing on traffic flow variability.

The subsequent sections of this paper are structured as follows. Section 2 provides a review of the literature concerning the application of RL methods in decision-making for last-mile logistics. Section 3 details the simulation-optimisation model utilised in this study. Section 4 describes the RL model, including notation, penalty calculation, and the bandit algorithm. Section 5 offers validation of the model, with subsection 5.2 delving into the discussion of experimental results. Finally, Section 6 presents the conclusions drawn from the study, highlights the limitations of the model, and set down avenues for future research.

2. Literature Review

Parking management strategies, including LZs management, have the capacity to affect the amount of traffic entering the city, reduce the capacity of roads, change the time of the trip, and in some cases, make the business move outside city centres (Shiftan & Burd-Eden, 2001). Unfortunately, for delivery vehicles (DVs) there is still a lack of quantitative studies related to parking demand and the impact of obstructing and illegal parking, preventing academics and practitioners from accurately defining the magnitude of the problem and designing appropriate solutions (Delaitre, 2009; Jaller, Holguín-Veras & Hodge, 2013). To overcome this issue, descriptive and prescriptive studies have been developed using a variety of methods. Descriptive approaches focus on understanding the capacity, availability, and occupation of LZs (Dezi, Dondi & Sangiorgi, 2010; Dablanc & Beziat, 2015; Malik et al., 2017). Prescriptive approaches are divided into the use of computer simulation for the evaluation of policies involving varying traffic conditions (Nourinejad et al., 2014; Dalla Chiara & Cheah, 2017; Iwan et al., 2018), illegal parking (Muñuzuri, Racero & Larrañeta, 2002; Delaître & Routhier, 2010; Jaller, Holguín-Veras & Hodge, 2013; Letnik et al., 2018) and enhanced law enforcement (Aiura & Taniguchi, 2005; Alho et al., 2018). Mathematical optimisation is used for defining the size, number, and location of LZs under deterministic and scenario-based conditions (Alho, De Abreu e Silva & De Sousa, 2014; Pinto, Golini & Lagorio, 2016; Tamayo, Gaudron & de La Fortelle, 2018). On the other hand, emerging technologies like sensors and GPS data are used to enable real-time

routing decisions (McLeod & Cherrett, 2011; Roca-Riu, Fernández & Estrada, 2015; Roca-Riu et al., 2017; Comi, Schiraldi & Buttarazzi, 2018; Yang, Roca-Riu & Menéndez, 2018). These solutions have focused on the optimisation of operational efficiency, the design, location, and management of LZs, and the reduction of freight-related externalities. However, there is still a need for decision support systems to help couriers make real-time parking decisions and how these decisions are affected by city freight dynamics.

Additionally, machine learning methods are usually divided into three categories (Naeem, Rizvi & Coronato, 2020): supervised learning, unsupervised learning, and reinforcement learning (RL). RL is the type of learning guided by a specific objective where an agent learns by interacting with an unknown environment using try-and-error. This environment typically changes due to the agent's actions and possibly other factors outside the agent's influence. When the agent perceives its environment, it collects information and decides which action to take so the accumulation of rewards is maximized (Nowé & Brys, 2016). Yan et al. (2022) provided a comprehensive review of the development and applications of RL techniques in logistics and supply chain management. They extended the review through a classification of previous research applications and provide an agenda for future directions. In a wider review, (Rof et al, 2023) focuses on RL applications to supply chain management.

RL has been widely applied to solve a variety of real-world problems, including robot control, shipping management problems, and production scheduling problems (Qiang & Zhongli, 2011). Only recently its practice has been intensified as a decision-support tool for last-mile logistics in transportation studies due to its strong performance and high applicability for decision-making (Ye et al., 2022; Yan et al., 2021). Applications span from traffic signal control (Wang, et al., 2020; Chu et al., 2019; Aragon-Gómez & Clempner, 2020), vehicle routing (Saravanan & Ganeshkumar, 2020; Zhao & Zhao, 2020), movement control (Passalis & Tefas, 2020; Guo et al., 2020) and traffic operations control (Wu, Kreidieh, Parvate, Vinitsky & Bayen, 2017). Although in a different setting, the search for charging stations in electric-vehicle related problems has a similar structure. Guillet et al. (2022) studied stochastic route search algorithms to make real-time decisions to minimise detours when looking for an available charging station. They modelled the system as a finite-horizon Markov decision process and presented a comprehensive framework that considers different problem variants, speedup techniques, and three solution algorithms: an exact labelling algorithm, a heuristic labelling algorithm, and a rollout algorithm. Results show that the proposed algorithm significantly decreases the expected time to find a free charging station while increasing the solution-quality robustness. A similar approach was taken by Guillet and Schiffer (2022), however improving over practical and theoretical approaches regarding driver coordination on the availability of charging stations, and the driver's intentions to visit the station. They modelled a multi-agent stochastic charging station search problem as a finite-horizon Markov decision process and introduce an online solution framework applicable to static and dynamic policies. Results show a system cost reduction of 26% and driver's search time savings of 23%. Basso et al. (2022) proposed an RL method to solve a dynamic vehicle routing problem (VRP) for electric vehicles. The goal is to minimize both power consumption and the risk of battery depletion by planning to recharge when needed. Stochastic customer requests and energy consumption are considered using Monte Carlo simulations based on energy consumption data from a real traffic model of Luxembourg City. Energy savings are achieved and at the same time, vehicles can make decisions in real time by predictively planning the route and ensuring that the vehicle does not run out of power while driving.

Several works have employed RL combining simulation and optimisation to solve both single-agent systems (SAS) and multi-agent systems (MAS). In SAS, a centralized entity makes all decisions and every agent acts as a remote slave, interacting with its environment. The results of all agent's interaction are then sent back to a single central processor. SAS applications to vehicle routing problems and vehicle path planning problems are presented by Bouhamed et al. (2019), Yu and Yu, (2019), Guo et al. (2020) and Zhao, Mao and Zhao (2020). In comparison, MAS systems differ from SAS in that the environment dynamics are not only influenced by the uncertainty of the environment itself but also by other agents' independent decisions. This makes MAS to have a dynamic behaviour. This is an inherent characteristic of the urban logistics environment where DV from many different companies interact, compete and sometimes cooperate to maximise their own rewards. Applications of MAS with RL are presented by Teo et al. (2012), who developed a MAS with road pricing to evaluate e-commerce logistics schemes for multiple stakeholders with the objective to manage the number of trucks in the city and the reduction of pollution levels. Results showed that cordon-based pricing is more effective than distance-based in reducing the pollution levels in the city centre. A later work by Teo et al. (2015) used an MAS and Q-learning to assess the effectiveness and viability of Urban Distribution Centres (UDC) from a tactical viewpoint. Results showed that UDCs have the capacity to relieve traffic congestion and negative environmental effects. An application to urban food distribution is presented by Chen et al. (2019) who studied a courier dispatching problem consisting of a set of couriers and pick-up requests with stochastic spatial and temporal arrival rates among urban regions. The model was formulated as a Markov Decision Process (MDP) with RL. Results from artificial and real-world data sets show that the solution approach achieves significant improvement over human dispatching policies. A similar problem

was faced by Jahanshahi et al. (2021) with additional variations including order rejection and courier repositioning. A case study is presented to Istanbul showing that the model proposed outperforms current algorithms with respect to both collected reward and the delivery times in a varying number of couriers.

A MAS with parking restrictions is proposed by Wangapisit et al. (2014). Using Q-learning the authors evaluate the effect of a Joint Delivery System (JDS), an UDC, and parking space restriction. Results show that the operating cost reduction and minimal environmental impact of implementing an UDC are encouraging. Using a similar approach, Tamagawa et al. (2010) presented a MAS that considers each stakeholder as an independent agent. Like Wangapisit et al. (2014), the authors used a combination of a VRP and Q-learning with multiple stakeholders to evaluate different logistic measures. Results show that the implementation of truck ban and discounting tolls did not compete against each other, but increased the environmental benefits. Firdausiyah et al. (2019) used adaptative dynamic programming-based RL with multi-agent simulation to model the behaviour of freight carriers and an UDC. An application is made to the city of Yokohama (Japan). Results showed the superiority of the model over more classical approaches such as MAS-Q-learning and proves that the learning itself is essential in the decision-making process of agents, especially when subject to a changing environment with multiple stakeholders. Yu et al. (2019) developed a deep RL-based neural combinatorial optimisation strategy to transform an online routing problem into a vehicle tour generation problem. The model is applied to a real-world network in the city of Cologne (Germany) with random pick-up and dropping points. Results showed that the proposed strategy can develop better vehicular tours compared to conventional mathematical programming methods with less computation time. Qin et al. (2021) proposed a dynamic task assignment method using a Multi-Agent System based on RL to solve the problem of controlling traffic signals in a complex urban transport network. In this approach, vehicles can make decisions in real time, reducing costs and obtaining higher service levels. Gupta, Ghosh & Dhara (2022) reduced the number of vehicles in operation and thereby the total cost of transportation. They presented an algorithm for fast and approximate solutions to the capacitated VRP with Time Windows using a combination of Multi-Agent Deep-RL and heuristics. Using numerical experiments, they manage to close the optimality gap and improve the computational time compared to Bono et al. (2020).

Although there is a vast literature on RL applied to last-mile logistics, current research does not consider operational decisions when DVs are making en-route decisions for searching unoccupied LZs. This work aims to fill this gap by using a combination of a simulation and optimisation model with a stochastic k-armed bandit that minimises the consequences (penalties) of making mid-route decisions that were not considered in the initial route planning stage.

3. Simulation-optimisation model

The proposed framework integrates a stochastic traffic microsimulation, an optimisation model, and a RL model. See Figure 1. The microsimulation consists of two layers; a lower layer describing the road network (entry and exit points, LZ locations, traffic demand, speed limits, intersections, and traffic light settings), and an upper layer that implements CA concepts to manage the agents, PVs and DVs, their sizes, speeds, motion, lane changing, routing behaviour, and delivery and parking decisions (for PVs only). Both PVs and DVs act within a multi agent environment and have an independent behaviour (non-cooperative). The optimisation model applies an evolutionary algorithm to solve the initial delivery route for each DV, which is a two-level (trucking and walking) VRP problem, and a commercial solver is used to re-optimise the DVs' mid-route decisions. Finally, the RL model is developed based on a stochastic k-armed bandit to support the decision-making process of DVs when a LZ along the initial delivery route is found unavailable. This section gives a brief description of the simulation and optimisation components and Section 4 focuses on the RL model. For the sake of self-containment here we describe the main features of the simulation-optimisation framework. For a detailed explanation of the microsimulation and optimisation models, the reader is referred to (Muriel et al., 2022).

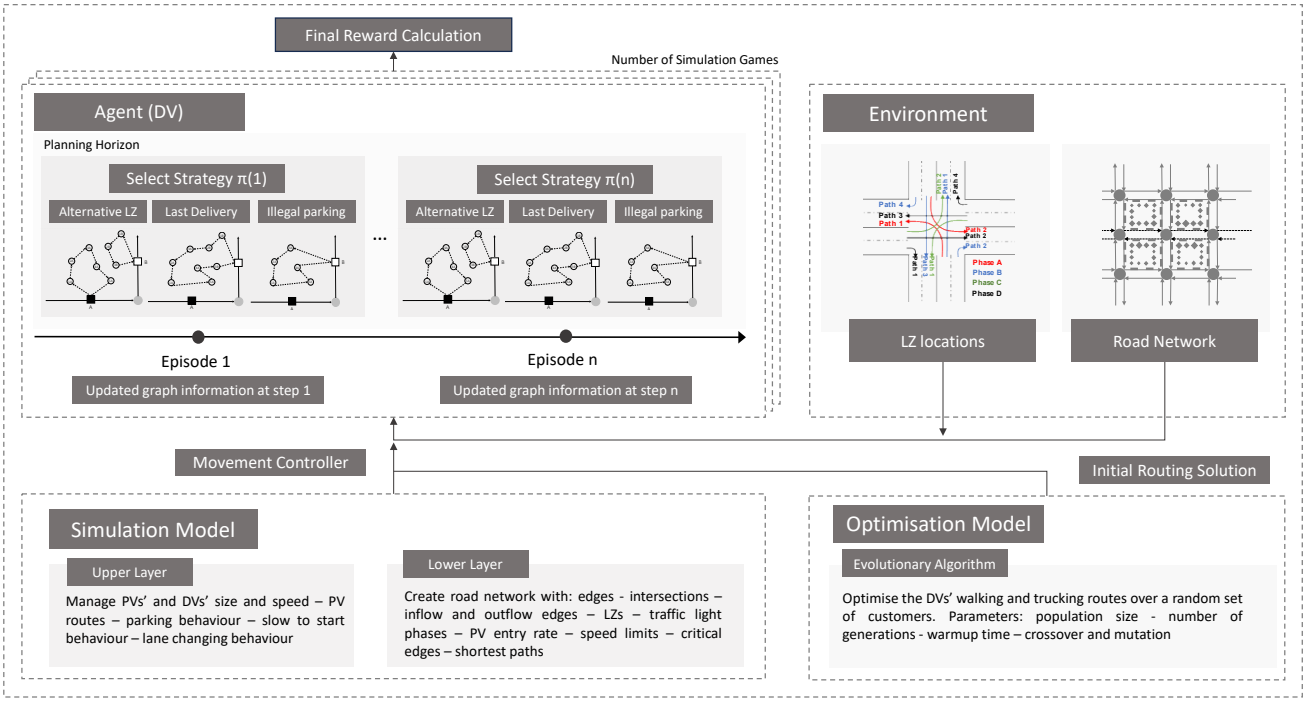


Figure 1. Schema of the simulation, optimisation and RL models

3.1 Simulation Model

The simulation model comprises a lower layer that manages the road network and an upper layer that controls the agents (PVs and DVs) behaviour. For a detailed description of the algorithms that govern the simulation process, the reader is refered to Muriel et al. (2022).

3.1.1 Lower Layer – Road Network

The graphical description of a bi-directional link and a signalised intersection of the road network is shown in Figure 2. The road network is represented as a connected digraph $G = (V, E)$, where V is the set of vertices and E the set of edges. The vertices represent the road intersections, while the edges represent the road network and each is composed of cells that can be occupied by vehicles. PVs and DVs can occupy several cells depending on their lengths. Cells can also be associated with a LZ for DVs to park in and its front and rear cells could be used for illegal parking. Figure 2 (left) shows an example of this structure for one edge and two vertices.

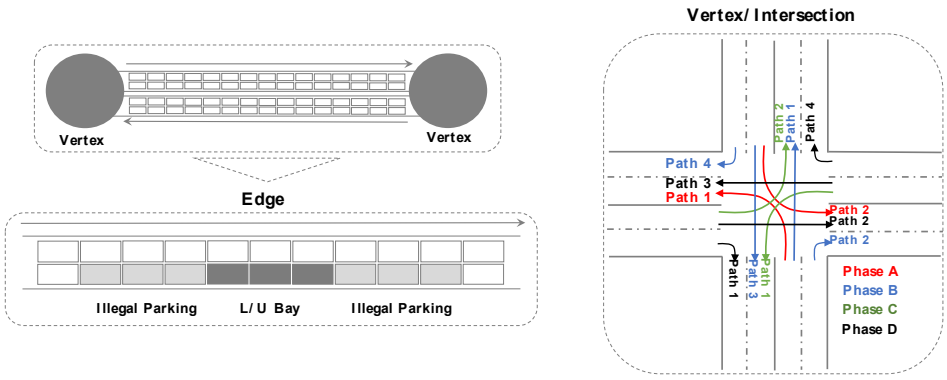


Figure 2. Example of a road link (edge) and a traffic intersection (vertex) with traffic signal settings. Taken from Muriel et al. (2022).

The road network has *inflow* and *outflow* edges at its boundaries. Inflow edges have a vehicle generator that inserts PVs and DVs into the network provided there are sufficient space to accommodate them. The insertion of PVs is controlled by a predefined *inflow rate*. The number of DVs that will enter the network during the simulation is fixed. If the inflow edge has no space to allocate a DV, it will enter a queue with a higher entry priority than PVs. PVs leave the network once

they arrive at one of the outflow edges, while DVs have to complete all their delivery requests before exiting the system via the outflow edges. To achieve realistic behaviour at the intersections, we introduce traffic signals with phases and paths shown in Figure 2 (right). In our model every vertex has four phases (A, B, C, D) that run sequentially. The traffic signal control strategy follows the *GreenWave* method (Wei et al., 2019). At any time, one vertex has only an *active* phase with a predefined phase time (green time), and only the paths belonging to the phase are active. A vehicle can cross the intersection only if its turning decision matches one of the active paths, and it will queue at the end of the edge, otherwise.

In most real-world road networks, traffic demand is heterogeneous. There are usually some streets that are more attractive and hence more congested than others. To model this heterogeneity, we define *critical edges* and introduce biased travel preferences. PVs are more likely to travel through the critical edges, which leads to higher traffic congestion for these roads. On the other hand, the route of a DV is specified by the sequence of LZs (and customers) that it needs to visit, and this sequence is determined by the optimisation model and could be altered following one of the delivery strategies.

3.1.2 Upper Layer – Agent Behaviour

Generally, the motion of PVs and DVs follows the Nagel-Schrekenberg (NaSch) model (Nagel & Schreckenberg, 1992), a variant of CA. While PVs have more aggressive driving that includes rapid accelerations, higher speeds and frequent lane changing, DVs tend to have more passive driving due mainly to their larger size and weight. In our work, the behaviour of the PV and DV agents was modelled based on the work by Long (2000). We adapt the variable acceleration function proposed by Rawat, Katiyar and Gupta (2012) and the lane-changing behaviour proposed by Zeng et al. (2016). The routing behaviour of DVs will be discussed in the following subsections.

3.2 Optimisation Model

The most important behaviour of the DVs is the optimisation of the delivery routes to complete a set of delivery requests. An example of the delivery route is illustrated in the left hand of Figure 3. The route consists of two levels: the first-level trips travelled by the DV, and the second-level trips travelled by the DV's driver (courier) on foot. The delivery route planning problem for each DV is an instance of the classic single truck and trailer routing problem with satellite depots (STTRPSD), where a single vehicle, based at a main depot, serves the demand of a set of customers reachable only by the truck without the trailer. The truck with the trailer mode in the STTRPSD corresponds to the DV driving mode in our study, the truck without the trailer mode corresponds to the walking mode, and the satellite depots are the LZs. The STTRPSD minimises the total distance travelled by the truck and the trailer (Villegas et al., 2010). The STTRPSD is gaining popularity for its usefulness in modelling two-level last-mile logistic decisions using urban consolidation centres, drone deliveries or, as in this case, park and loop deliveries. Integer programming formulations of the STTRPSD have been made by Villegas et al. (2010) and Belenguer et al. (2016) and with a similar structure by Martinez-Sykora et al. (2020), Reed, Campbell and Thomas (2021), Thompson and Zhang (2018) and Muñuzuri et al. (2012). Literature reviews on the topic are made by Cuda, Guastaroba and Speranza (2015) and Slujik et al. (2023). The complete formulation of the STTRPSD is presented in Appendix A.

The STTRPSD is an NP-hard problem (Villegas et al., 2010), and the preferred solution methods for this and other related problems are metaheuristics (c.f., Lamas-Fernandez, et al, 2023; Cavagnini et al, 2023; Accorsi and Vigo, 2020) . In this work, every DV agent needs to solve an independent STTRPSD. We applied the evolutionary algorithm (EA) developed in (Muriel et al, 2022) to solve the problem. Evolutionary algorithms have been used to solve the capacitated vehicle routing problem and several of its extension successfully (Potvin, 2009), Vidal et al., 2013). Therefore, we follow this approach as it has proved to be useful in the solution of complex VRPs while keeping a relatively simple design. The EA representation uses two-levels aimed at representing the solution of the problem. A list of LZ where the DV is going to park in the first-level trip $\Pi = \{E_{inflow}, \pi_1, \pi_2 \dots, \pi_j, \dots, E_{outflow}\}$ and then second-level trips visiting the customers assigned to each LZ (tsp_{π_j} for $j = 1 \dots |\Pi| - 1$). The fitness function of a solution (σ) of the EA used to optimise the DV routes comprises two terms considering the different speeds of the vehicle traveling between LZs and the walking tours to the customers. The first term sums the trucking times of the DVs in the first-level trip ($TTime(\sigma) = \sum_{j=0}^{j=|\Pi|-1} TTime(\pi_j, \pi_{j+1})$ and the second one sums the walking times for the delivery following a travelling salesman problem (TSP) from each LZ ($WTime(\sigma) = \sum_{j=1}^{j=|\Pi|-1} WTime(tsp_{\pi_j})$. Since the STTRPSD usually optimises the total distance, we consider alternatively a standard walking and driving speed to get the total time of the route for both terms. The left-hand side of Figure 4 depicts, the representation of the EA solution.

Initially, the trips to visit the customers once parked in a given LZ are found using a nearest neighbour heuristic. As suggested by Ahuja and Orlin (1997), the EA is equipped with a greedy randomised construction for the initialisation to

guarantee a fast convergence without compromising the simulation time. To enhance its computational performance, the EA includes elitism and a biased crossover selection, where new solutions are obtained by applying crossover with the fittest individual with high probability. The mutation process uses three different mutation operator (one swapping customers between second level trips, and two other aimed at swapping or merging the stops at the LZ's in the first level trip). Finally, as a solution improvement strategy, when the DV's has to replan the trip due to LZ unavailability, the second-level trips are re-optimised using the Miller-Tucker-Zemlim (MTZ) (Miller, Tucker & Zemlin, 1960) formulation of the TSP via a commercial optimiser. For further details on the EA design the interested reader is referred to Muriel et al (2022).

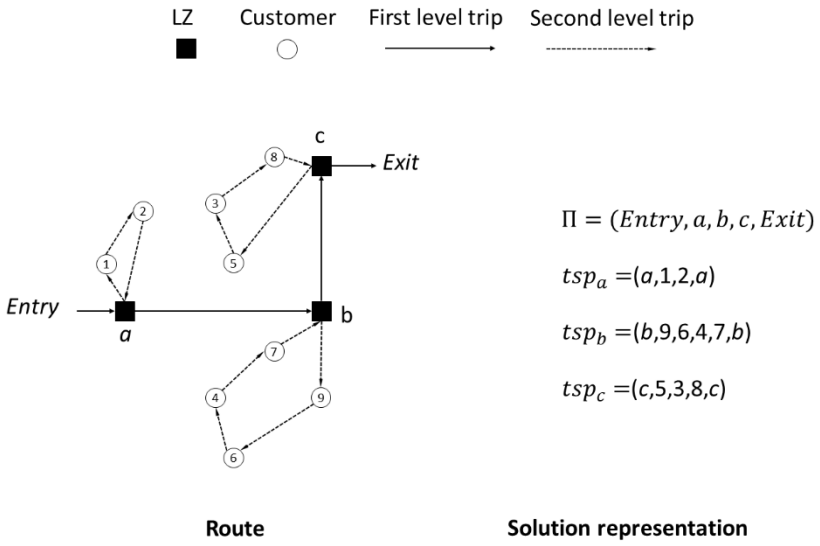


Figure 3. Solution representation of STTRPSD within the EA

An important feature of the proposed model is the realistic assumption that DVs do not cooperate nor have a centralised control, therefore solving a multi-vehicle problem (Martinez-Sykora et al. 2020; Reed, Campbell and Thomas , 2021) is not a viable solution strategy to model the decisions of the DVs when planning their routes. Rather, they solve and re-optimise their parking decisions independently during the simulation process using the reinforcement learning model that follows.

4. Reinforcement Learning Model

As mentioned before, this study embeds reinforcement learning into the simulation-optimisation model to help DVs choose delivery strategies when en-route decisions are needed. En-route decisions comprise the set of reasonable actions taken by a DV that deviate from the initial optimal delivery plan (route optimisation). These decisions reflect the different challenges faced by DVs on CBD areas (congestion, curfews, road closures, LZ competition, etc). In this section, we will briefly introduce the RL model, discuss the configuration of the penalty function, and finally explain the k-armed bandit algorithm developed to solve the RL model. For an in-depth introduction to RL, we recommend referring to Naeem, Rizvi, and Coronato (2020).

In our model, each DV is a learning agent (decision maker) that interacts with its environment (the road network). The constant interaction between the DV by selecting new actions and the environment responding to these actions and presenting new situations is depicted Figure 5. Every state in $\mathcal{S}$ must include information about all past agent-environment interactions that define the future penalties (called the Markov property).

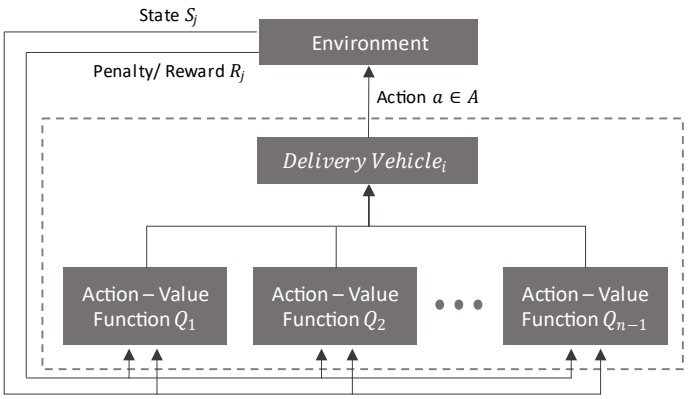


Figure 4. Agent-environment interaction in an MDP

The DV and the environment interact sequentially over m games and n episodes. A game is one simulation run with an episode representing the decision point where the DV finds a LZ unavailable and needs to take an action $a_j \in \mathcal{A}$ that is fed to the environment, receiving a penalty $r_j \in \mathbb{R}$ from distribution P_{a_j}. The number of actions taken by an agent is a stochastic variable since the number of times a particular DV finds a LZ occupied is unknown. The interaction between the agent and the environment induces a probability measure on the sequence of outcomes $a_1, r_1, a_2, r_2, \dots, a_n, r_n$ that should satisfy the following assumptions (Lattimore & Szepesvári, 2020):

(a) The conditional distribution of the penalty r_j given $a_1, r_1, \dots, a_{j-1}, r_{j-1}, a_j$, is P_{a_j}, which captures the intuition that the environment samples r_j from P_{a_j} in episode j.

(b) The conditional law of action a_j given $a_1, r_1, \dots, a_{j-1}, r_{j-1}$ is $\kappa_j(.|a_1, r_1, \dots, a_{j-1}, r_{j-1})$, where $\kappa_1, \kappa_2, \dots$ is a sequence of probability kernels that characterise the agent. The most important element of this assumption is that the agent cannot use future observations in current decisions.

If $r_{n|a}$ denotes the penalty received after the nth selection of a particular action a, let Q_n denote the estimate of its action value after $n-1$ actions have been selected. Due to the non-stationary nature of the problem (penalty probabilities change over time) it seems sensible to give a higher weight to recent penalties than to long-past penalties. For this, the estimate of the value action can be calculated as an exponential recency-weighted average using a step-size parameter $\beta \in (0,1]$, as shown in Equation 15 (Sutton & Barto, 2018):

$$
\begin{aligned}
&Q_{n+1} = Q_n + \beta\left[r_{n|a} - Q_n\right] \\
&= \beta r_{n|a} + (1-\beta)Q_n \\
&= \beta r_{n|a} + (1-\beta)[\beta r_{n-1|a} + (1-\beta)Q_{n-1}] \\
&= \beta r_{n|a} + (1-\beta)\beta r_{n-1|a} + (1-\beta)^2 Q_{n-1} \\
&= \beta r_{n|a} + (1-\beta)\beta r_{n-1|a} + (1-\beta)^2\beta r_{n-2|a} + \cdots + (1-\beta)^{n-1}\beta r_{1|a} + (1-\beta)^n Q_1 \\
&= (1-\beta)^n Q_1 + \sum_{j=1}^{n} \beta(1-\beta)^{n-j} r_{j|a}
\end{aligned}
\tag{15}
$$

Since $0 \leq \beta \leq 1$, the weight given to r_j decreases as the number of intervening penalties decreases. It is important to note the similarity of this equation with the Gradient Ascent method (Luke, 2013) which is the foundation for function optimisation ($x \leftarrow x - \beta f'(x)$ for a unidimensional function). The main difference is that what corresponds to the function's slope $f'(x)$, is now the difference between the new penalty and the current function value $\left[r_{n|a} - Q_n\right]$, that ultimately drives the search.

4.1.1 DV Route Update and Penalty Calculation

When a DV agent arrives at a LZ where it would like to park but finds the LZ unavailable, it will take one of three actions $\mathcal{A} = \{Illegal\ Parking, Alternative\ LZ, Last\ Delivery\}$. The penalty $r_{j|a}$ received by the agent after taking action $a_j \in \mathcal{A}$ is calculated based on the time increment for following the parking strategy compared to the delivery time of the current route. Therefore, this quantity varies on every action a_j. The description of the route updates and corresponding penalties follow. As a summary, Table 1 presents the detailed calculation of the penalty generated by a given action as well as the changes in the first- and second-level trips derived from taking each one of the available actions.

Illegal Parking

If taking the *Illegal Parking* action, when a DV finds the designated LZ d unavailable, it first performs one block circle before attempting to park illegally at the rear or front of the LZ. This strategy aims for minimising cruising and walking times but likely causes/increases traffic congestion and parking fines. Note also that this action does not modify the first-level trip nor the TSP for the delivery from d. Therefore, the penalty is calculated based on the probability for the DV to get a fine while parking illegally, p_{Fine}, or in the worst case getting towed away.

For this, we defined the parameter *FineValue* as the penalty received by the DV when p_{IP} (a random probability) is less than or equal than the probability p_{Fine}. *FineValue* is a time-converted penalty that is set to be significantly higher than the one obtained with *Alternative LZ* or *Last Delivery* strategies to represent the risk-taker behaviour of DVs when parking illegally. The probability p_{Fine} is a time-increasing function to simulate the fact that the longer the vehicle remains parked, the more likely it gets fined. To capture this behaviour, we used the exponential function depicted in Equation (16) of Table 1 and Figure 5, where *DelTime* is the delivery time (the dwell time) in the LZ and *MaxDelTime* is the 98th delivery time percentile. To represent the effect of different enforcement levels, we varied parameters ω and θ as shown in Figure 7. We consider three levels of enforcement. On one end, a high enforcement level translates in high probabilities for DV of getting a fine in short parking times. In the other end, a low enforcement level implies that DVs need to significantly extend their stay in order to get a fine. Finally, we evaluate a medium enforcement level where there is a relatively 'fair' probability of getting a fine depending on the length of the stay. Delivery time can vary from a few minutes up to a maximum of 15 minutes (900 seconds).

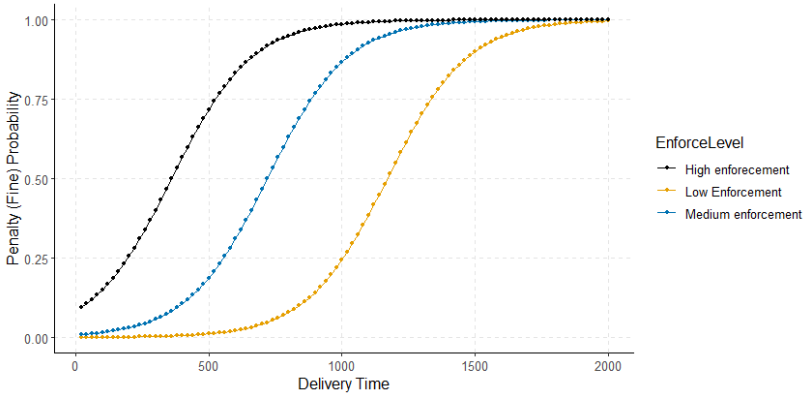


Figure 5. Illegal parking function for generating fine probabilities

Alternative LZ

Figure 6 show a representation of the *Alternative LZ* action. When a DV finds the LZ d occupied, it first starts cruising with the shortest return loop. If LZ d is still unavailable after one loop, it modifies the delivery by finding the shortest path to the closest LZ d'. At this point, DV will receive one of two possible penalties. *(i)* If the closest LZ d' is already on the delivery route, the customers from both LZs are joined together (Figure 6 left) ($tsp'_{d'} = tsp_d \cup tsp_{d'}$), and the new walking tour is re-optimised by solving a new TSP using the commercial solver. The penalty from this option is the difference between the new walking ($WTime(tsp'_d)$) and the previous walking times initially assigned to d ($WTime(tsp_d)$) and d' ($WTime(tsp_{d'})$) (See equation 17 of Table 1). *(ii)* On the other hand, If the closest LZ is not on the delivery route

(Figure 6 right), the penalty is calculated as the driving time from LZ d to LZ d' ($TTime(d, d')$) plus the time to make the walking tour and serve the customers from LZ d' ($WTime\left(tsp_{d'}\right)$), subtracting the original time to serve the customer from LZ d $WTime(tsp_d)$) (See equation 18 of Table 1). Nonetheless, if by the time the DV reaches the alter native LZ d'it is unavailable, the DV adopts the illegal parking strategy.

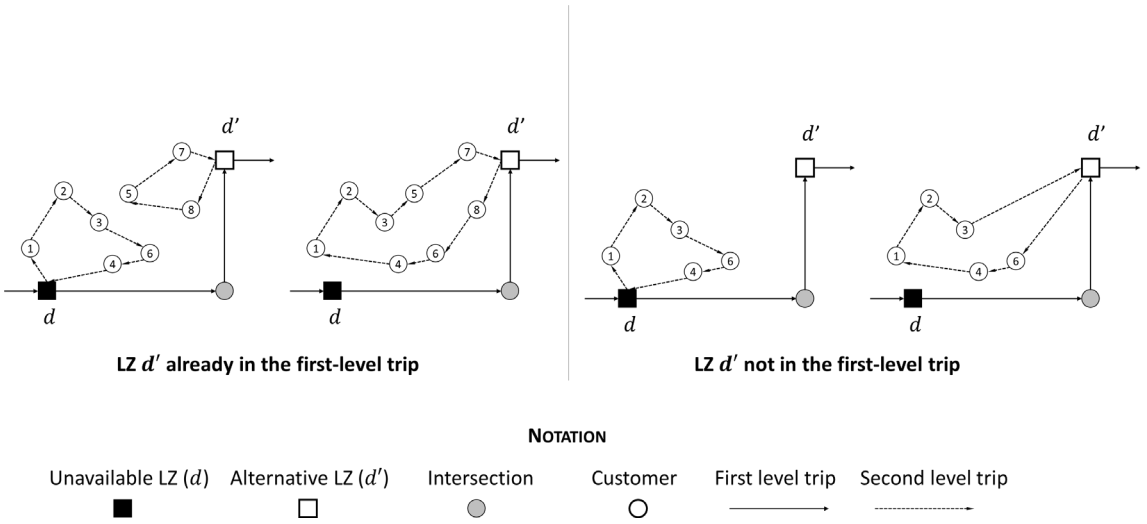


Figure 6. Representation of the *Alternative LZ* action

Last Delivery

Finally, in the last delivery strategy, when a DV finds the designated LZ d unavailable it puts the delivery request(s) for this LZ to the end of the route the first-level trip. This implies a change in the first-level trip (Π). Then the penalty is calculated as the driving time needed for returning from the last LZ d_{last} to the current LZ d ($TTime(d_{last}, d)$), plus the travel time from the current LZ d to the exit edge E_{exit} ($TTime(d, E_{exit})$), less the driving time that was originally assigned from the last LZ d_{last} to the exit edge E_{exit} ($TTime(d_{last}, E_{exit})$). Figure 7 and equation (19) of Table 1 represents this strategy.

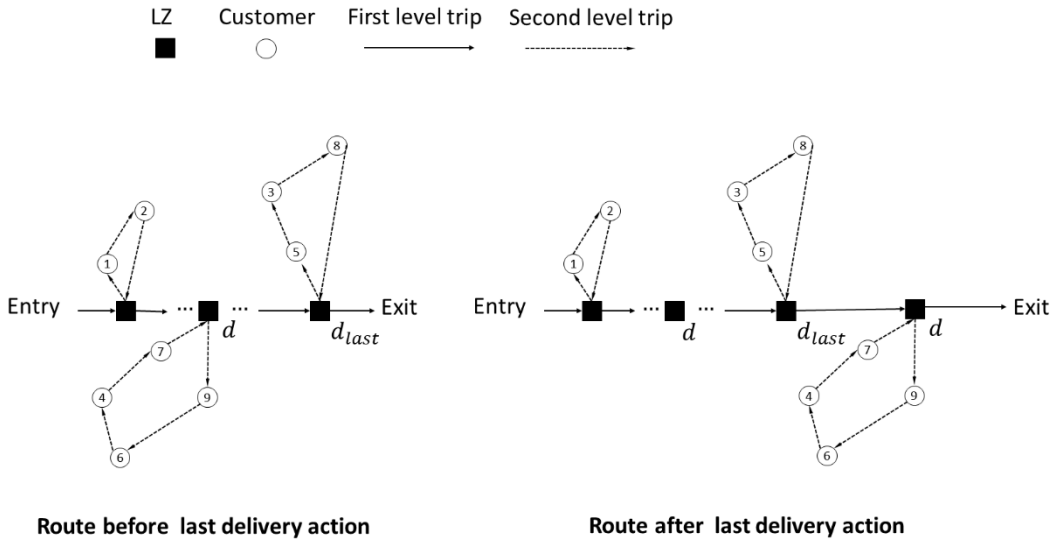


Figure 7. Representation of the Alternative LZ action

As described above, depending on the action taken by a DV when the planed LZ d is unavailable the DV changes its first-level trip and parks in a different LZ d' (if illegal parking is not chosen). If needed the re-optimisation of the second-level trip is also performed. Table 1 summarizes the behaviour of the vehicles for each one of the actions. The first row of the table presents the current route and the case when LZ d is available upon arrival of the DV, that does not change the route and does not require any action. The next rows describe the changes in the route and the corresponding penalties

Table 1. Summary of route modification and corresponding penalties for each available action

Action	New LZ	First-level trip	Reoptimised second level trip	Penalty	
d is available (no action)	None	$\Pi = \{E_{inflow}, \pi_1, \pi_2 \dots, d, \dots, d_{last}, E_{outflow}\}$	No	0	
$Illegal\ Parking$	None	$\Pi = \{E_{inflow}, \pi_1, \pi_2 \dots, d, \dots, d_{last}, E_{outflow}\}$	No	$r_{j\|Illegal\ Parking} = \begin{cases} FineValue, if\ p_{IP} \le p_{Fine} = \dfrac{1}{1+\exp\left[\dfrac{\omega(\theta * MaxDelTime - DelTime)}{MaxDelTime}\right]} \\ 0, otherwise \end{cases}$	(16)
$Alternative\ LZ$	$\boldsymbol{d'} \in \Pi$	$\Pi = \{E_{inflow}, \pi_1, \pi_2 \dots, d, d', \dots, d_{last}, E_{outflow}\}$	Yes	$r_{j\|AlternativeLZ} = WTime(tsp'_{d'}) - ((WTime(tsp_d) + WTime(tsp_{d'}))$	(17)
$Alternative\ LZ$	$\boldsymbol{d'} \notin \Pi$	$\Pi = \{E_{inflow}, \pi_1, \pi_2 \dots, d, d', \dots, d_{last}, E_{outflow}\}$	No	$r_{j\|AlternativeLZ} = TTime(d, d') + WTime(tsp_{d'}) - WTime(tsp_d)$	(18)
$Last\ Delivery$	None	$\Pi = \{E_{inflow}, \pi_1, \pi_2 \dots, d, \dots, d_{last}, d, E_{outflow}\}$	No	$r_{j\|LastDelivery} = TTime(d_{last}, d) + TTime(d, E_{exit}) - TTime(d_{last}, E_{exit})$	(19)

4.1.2 Bandit Algorithm

We chose a stateless approach to model DVs' parking decisions since it is not possible to know the occupancy status of all the LZs in the entire system. A contextual bandit (Bouneffouf et al, 2020) is a possible alternative approach, but a centralised decision-making framework with full information of all DVs and LZs does not seem realistic as stated above. Additionally, if a contextual bandit is used, the context vector will require a position to track the occupation of each LZ. Therefore, the context space will be of size $2^{|B|}$ (where $|B|$ is the cardinality of the set of LZs) leading to a combinatorial explosion of the learning process. Alternatively, tracking the specific LZ in which the actions are taken and the next LZ to visit after taking an action could be an option of a possible state variable for the DVs. Using the LZ as state variable will enlarge the state-action space to be learned to a size equal to $|\mathcal{A}| \times |\mathrm{B}|$. Under this approach, each time a DV finds a LZ unavailable it will choose the action to perform, and therefore the next LZ to visit in the first level route. Note, however, that this approach completely ignores the initial planning of the DV's route made with the EA leading to a dynamic routing of the vehicles with a combinatorial explosion of the action space. As pointed out by Hildebrandt et al (2023), these detailed route-based state-action models are computationally expensive and requires a large number of observations to converge. For these reasons, we preferred the simple, yet effective, modelling approach that k-armed bandits offer as an initial step to incorporate RL-driven decisions of DVs into the simulation-optimisation framework proposed in Muriel et al (2023). Nonetheless, our multi-agent RL framework, in which each DV is modelled as an independent agent, can be seen as a state restriction that improves the efficiency of the learning process (Hildebrandt et al, 2023).

In this context, k-armed bandits represent a specific category of RL problems characterized by a single state, potentially multiple actions, and a unique feature where the learner doesn't require forward planning. In k-armed bandits, the agent's objective is to find an optimal control principle that maximises its rewards, or in our case minimises its penalties, by trial-and-error iterations with its environment (Yu et al., 2021). When an agent finds an action that gives repetitive high rewards (or low penalties), it faces two possible options: exploiting that action or continuing exploring the search space. The former case is a good example of a myopic or greedy algorithm, which takes the risk of getting stuck in a local optimum by favouring short-term rewards over possible long-term benefits. In the latter, an agent takes the risk of continuing to explore the space by observing rewards from actions that might not have been so profitable in the short term but could (hopefully) give higher long-term rewards. The right combination of exploitation and exploration has a direct influence on the algorithm convergence and its capacity to reach a global optimum given a limited number of episodes. Several strategies are used to find the right exploitation-exploration balance, including the widely used greedy and ε-greedy methods. While greedy methods spend no time at all sampling inferior actions, the ε-greedy methods behave greedily most of the time but, occasionally, select a random action (with some probability ε) among all actions independently of the action-value estimates (Sutton & Barto, 2018). While convenient, the greedy and ε-greedy methods use monotone learning rates that inevitably force the agent to either spend too much time on drawing suboptimal actions or completely fail to identify the optimal one (Cesa-Bianchi et al., 2017).

To avoid this problem, we use a similar approach to the Metropolis algorithm in the Simulated Annealing metaheuristic (Luke, 2013) known as *Boltzmann exploration*. We start the algorithm with a high probability of exploration and then make it become "greedier", as more knowledge is gathered about the true action values. One major advantage of ε-greedy methods is that when the number of episodes increases, every action is sampled an infinite number of times, ensuring that the estimated action value Q_j converges to its true action value q_{j*}. We adopt the function proposed by Šemrov et al. (2016) to compute the probability ε as shown in Equation (20) and Figure 8.

$$\varepsilon = \frac{P_{explor}}{1 + \exp\left[\frac{10(\mathrm{NA} - 0.4 * \mathrm{NM})}{NM}\right]} \tag{20}$$

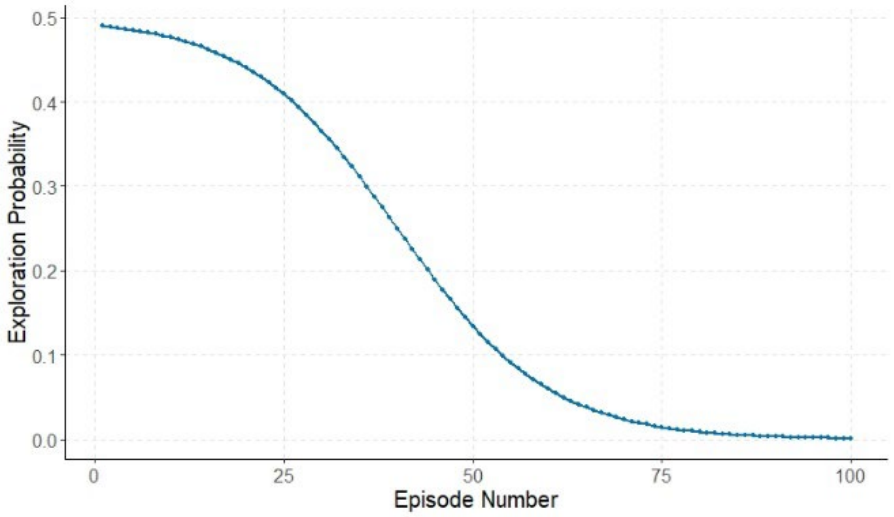


Figure 8. Function for generating the exploration/exploitation probability

NM is the total number of simulations (games), NA is the total number of actions taken so far and P_{explor} is the initial exploration probability (equal to 0.5 in Figure 8). The detailed algorithm of the k-armed Bandit is given in Algorithm 1. When a DV i finds a LZ d unavailable it generates the exploration probability ε (line 5, Algorithm 1) and takes a greedy action with probability 1- ε (action with minimum state value) or a random action with probability ε (line 6, Algorithm 1). After calculating the penalty r_j for the chosen action (line 7, , Algorithm 1), it updates the DV route, number of actions NA and the action value Q_j (lines 8-10, Algorithm 1).

Algorithm 1 k-Armed Bandit Algorithm

Parameters: $action\ set\ A, number\ of\ games\ M, probability\ \varepsilon, step\ size\ \beta$
Initialise: $action - value\ table\ Q_{ja} \leftarrow 0\ and\ set\ NA \leftarrow 0\ , episode\ j = 0$

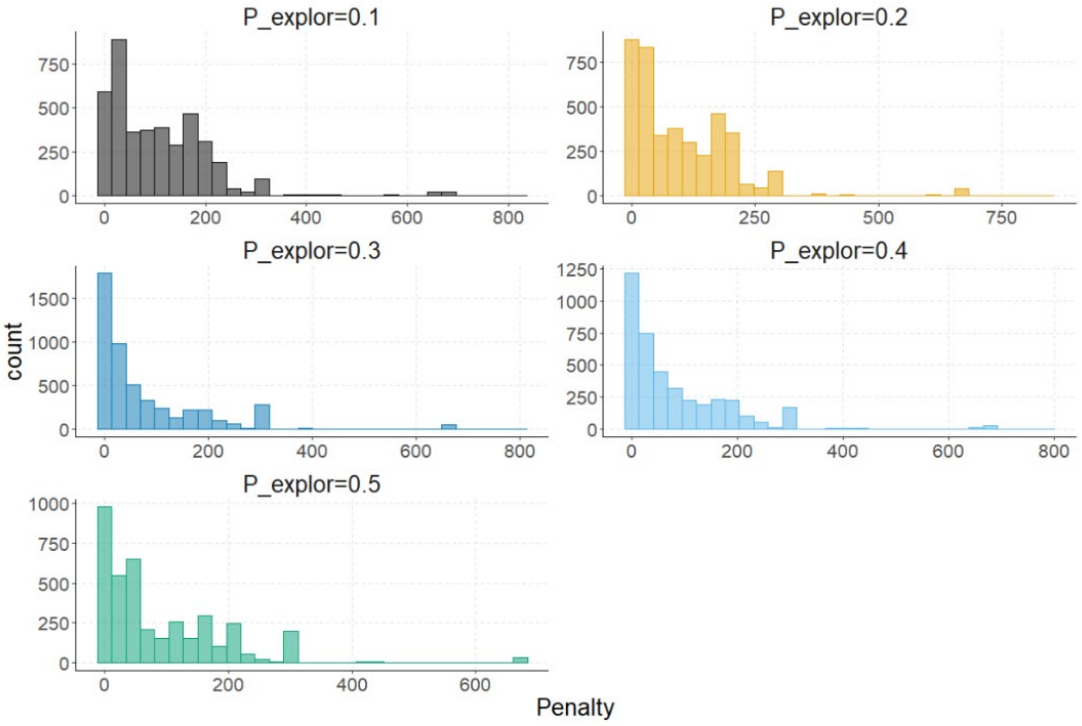
1: ***for*** $\mathbf{m \leftarrow 1\ to\ M}$ ***do***
2: ***for*** $each\ DV$ ***do***
3: ***for*** $each\ LZ\ d\ on\ first - level\ trip\ \Pi$ **do**
4: ***if*** $LZ\ is\ occupied$
5: $Generate\ \varepsilon\ according\ to\ Equation\ (19)$
6: $Take\ action\ a \in \mathcal{A} \leftarrow \begin{cases} argmin_a\ Q_{ja}\ with\ probability\ 1-\varepsilon \\ random\ action\ with\ probability\ \varepsilon \end{cases}$
7: $Calculate\ the\ penalty\ r_{ja} - According\ to\ action\ a\ (See\ Table\ 1)$
8 $Update\ \Pi\ and\ Second - level\ trips\ according\ to\ a\ (See\ Table\ 1)$
9: $NA \leftarrow NA + 1$
10: $Q_{(j+1)} \leftarrow Q_j + \beta[r_{ja} - Q_j]$
11: $j \leftarrow j + 1$
12: ***next*** d
13: ***next*** DV
14: ***next*** m
15: ***return*** $Action - value\ table$

5. Computational experiments

In this section we first validate and calibrate the RL model to finetune the key parameters that control the k-Armed-Bandit Algorithm. Once calibrated, we use the model to analyse the effect of different enforcement levels and also the impact of DV parking decisions on traffic network performance.

5.1 Model Validation and Parameter Calibration

We start calibrating the algorithm parameters by running 500 simulations (games) with 50 DVs per simulation. To find the effect of the exploration-exploitation parameter P_{explor}, we fixed the discount factor $\beta = 0.3$. Figure 9 shows the Probability (top) and the Cumulative Density (bottom) Functions (PDF, CDF) for the penalty value r_j, respectively. In the CDF the value of P_{explor} equal to 0.3 shows a positive skew with the highest number of lower penalties. Similarly, the CDF shows that around 75% of values have a penalty under 100-time units, compared to a P_{explor} of 0.1 that has only 55% of values with a penalty under 100-time units. Therefore, $P_{explor} = 0.3$ is an adequate value for this parameter.



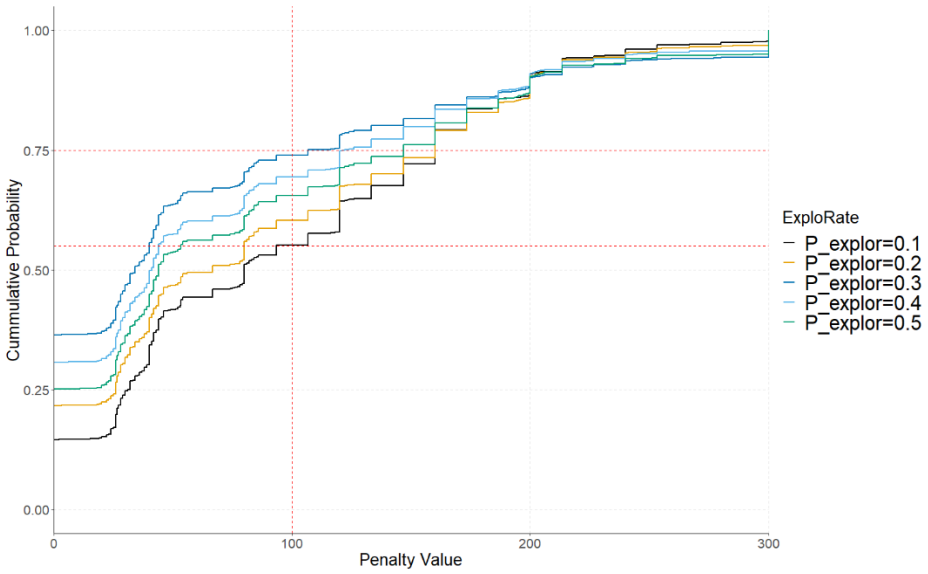
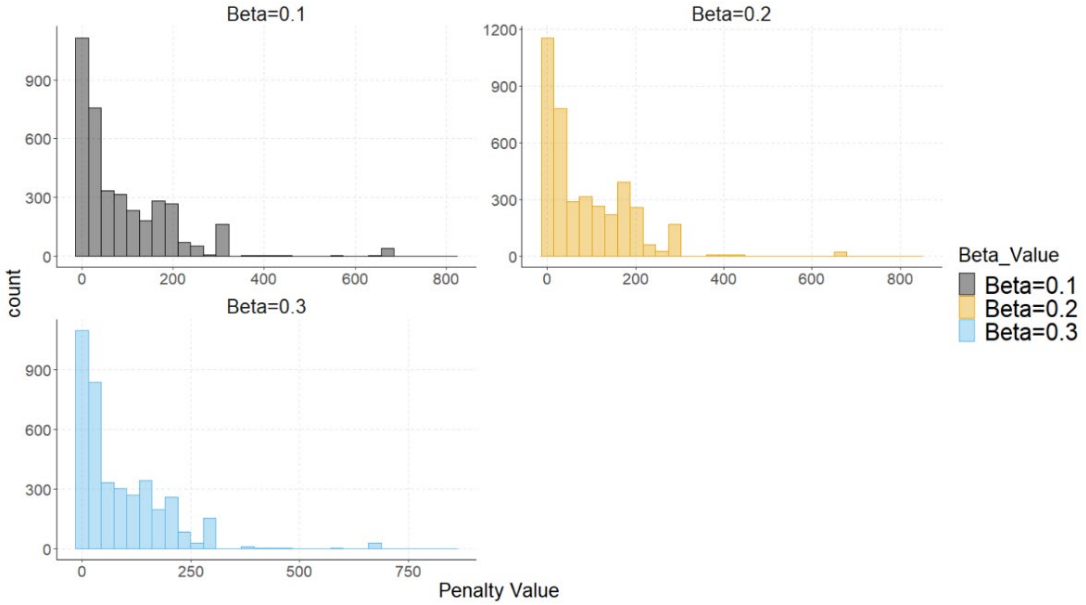


Figure 9. PDF and CDF of the penalty function with different exploration rates (P_explor=P_{explor})

To calibrate the value of β we made 500 experiments from 0.1 to 0.3 fixing P_{explor} at 0.3 and using Equation (15). The PDF (Figure 10 top) shows that although the three strategies have similar distribution functions, β equal to 0.1 shows the highest probability of getting low penalty values. However, this effect is not significant in the long term as can be observed in the CDF of Figure 10 (bottom). It is important to highlight that the penalty function for the *Illegal Parking* strategy might add noise to this calculation if long deliveries (with a high probability of being fined) are present at the end of the DV's route where the algorithm is "greedier" than at initial stages.



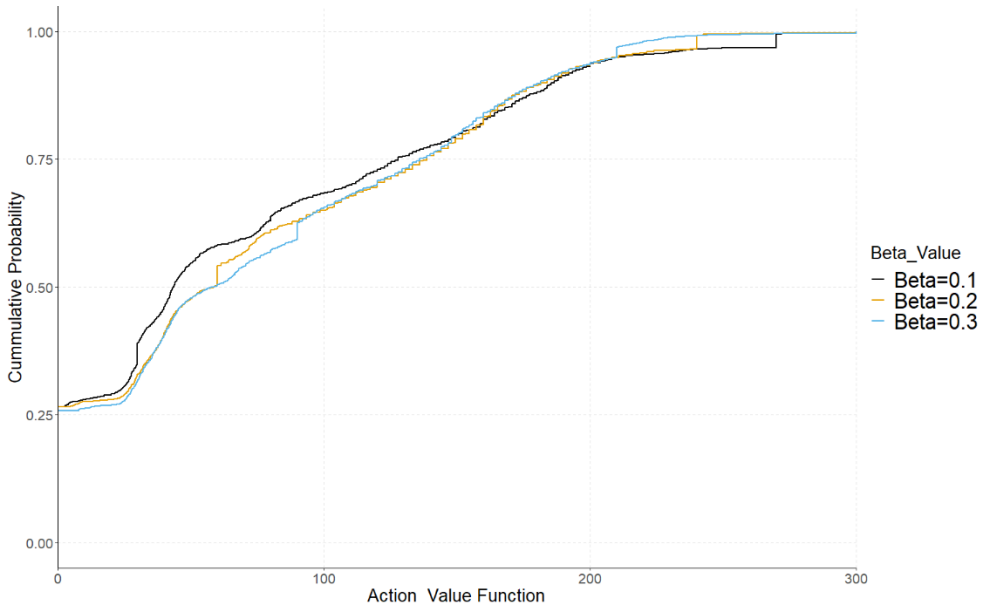


Figure 10. PDF of the penalty value and CDF of the action value function ($Beta = \beta$)

Figure 11 shows the action value function for three DVs comparing the proposed ε-greedy approach (using $P_{explor} = 0.3$ and $\beta = 0.1$) with a completely greedy method and a random approach. The greedy approach starts with the same behaviour as the ε-greedy but makes only a small number of actions before getting trapped in an action that gave initially the lowest penalties even if in the long term this strategy does not offer the best performance. The DVs in the middle column of figure 11 show this behaviour. For instance, DO40 and D047 behaviour shows a vehicle that does not get a fine in the initial episodes despite parking illegally, and therefore adopts this action for the entire simulation. However, in future parking decisions the DVs got fined for parking illegally obtaining very high penalties. On the other hand, the random approach always takes a random action regardless of the penalty value, therefore achieving the worst results. The DVs in the left side of Figure 11 illustrate this behaviour. Although the ε-greedy approach starts exploring the environment getting higher penalties at the beginning when compared to greedy and random approaches, it soon "learns" and exploits the best strategy, always choosing the minimum action value and occasionally taking a random action to avoid stagnation on local optima (as explained in Section 3.3.2).

Figure 12 presents the average aggregates for action values across all DVs under each approach, accompanied by their respective standard deviations. While accurately estimating convergence and optimality gap in this problem proves challenging due to the significant variability of penalty functions, the swift convergence towards lower state action values and the narrower confidence intervals observed with the ε-greedy approach unmistakably demonstrate the advantages of employing Boltzmann exploration.

The tendency of the ε-greedy approach to incur higher penalties in the initial episodes can be ascribed to the exploration cost associated with higher P_{explor} values. However, this cost diminishes notably in subsequent episodes with lower P_{explor} values, particularly after the algorithm 'learns' the strategy that yields superior results. Despite the random and greedy approaches yielding comparable state action values, it's noteworthy that the greedy approach exhibits narrower confidence intervals in Figure 12.

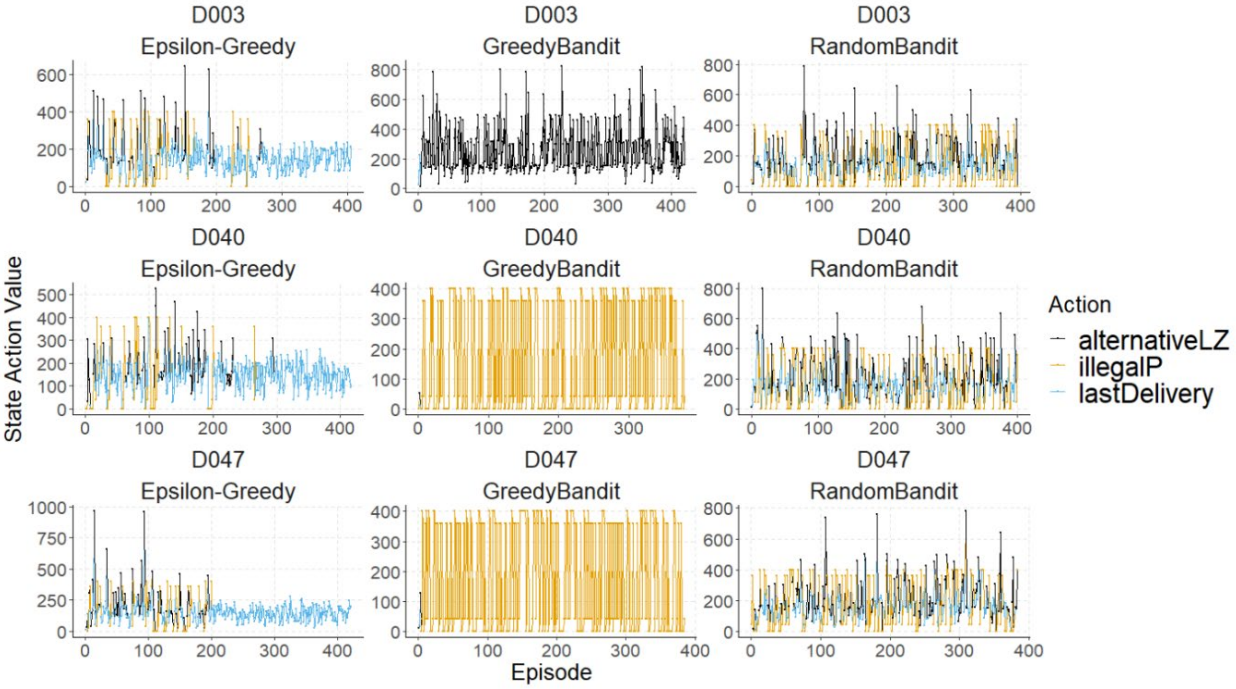


Figure 11. Action value time series for three DVs (D003, D040, and D047) (Epsilon=ε)

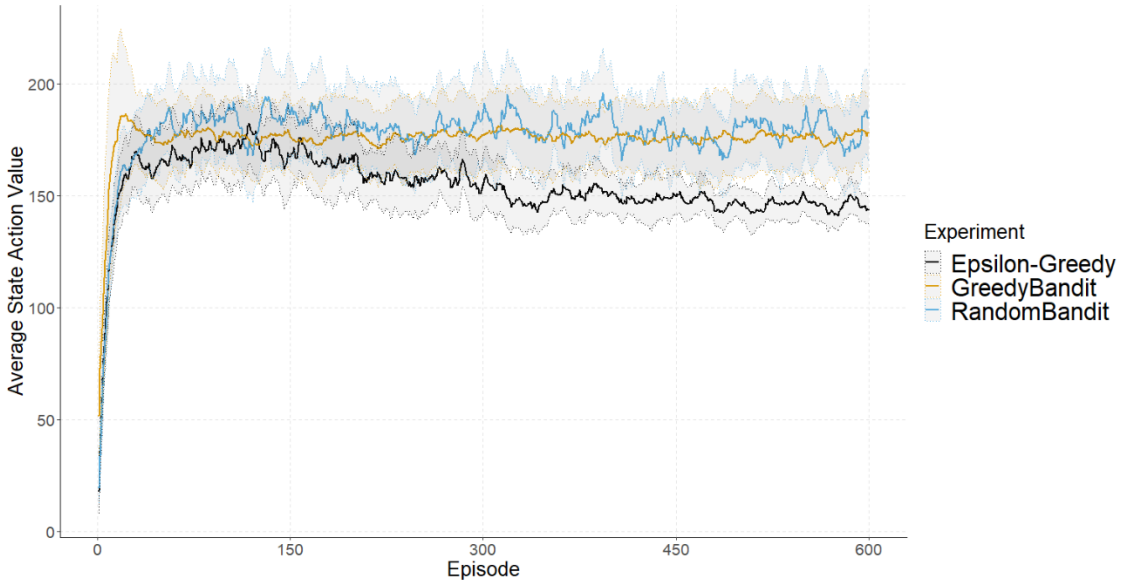


Figure 12. Action value averages for all DVs with standard deviation shown as shadow marks (Epsilon=ε)

.

5.2 Results

We use the lattice representation of the Melbourne CBD. A key characteristic that differentiates our model from other applications, is the fact that the City of Melbourne (the local council of the Melbourne central city areas) enforces DVs to

use specific LZs to park, making it impossible to serve customers by pure driving tours. The network has a total of 15 vertices, 76 edges with 12 critical streets, and 48 demand points (each point can be interpreted as a cluster of customers, for example, a commercial building). The LZs are equally distributed in the network with each road having 2 LZs located in the middle of the segment with an additional 4 illegal parking areas. Every DV keeps a record of its actions and their penalties for every LZ visited that was unavailable; not only during each simulation (game), but also over the whole set of the simulations $N(M)$. Hence, the value of ε is calculated over all NM games and NA actions taken. To evaluate the proposed model, we ran 1000 simulations with $P_{explor} = 0.3$ and $\beta = 0.1$. Figure 13 shows the penalty distribution for the three strategies (actions). The *Last Delivery* strategy has a usage percentage over all the DVs of 46%, followed by A*lternative LZ* with 29% and *Illegal Parking* with 25%. Although the *Alternative LZ* strategy has a higher probability of getting lower penalty values, it has a tail that goes up to 1200 seconds. Instead, the *Last Delivery* strategy has a central tendency with penalties going up to 300 seconds. It seems that the DVs are better off by choosing the Last Delivery strategy despite being "penalised" with the additional driving time, rather than choosing *Alternative LZ* that has a small probability to have big penalties.

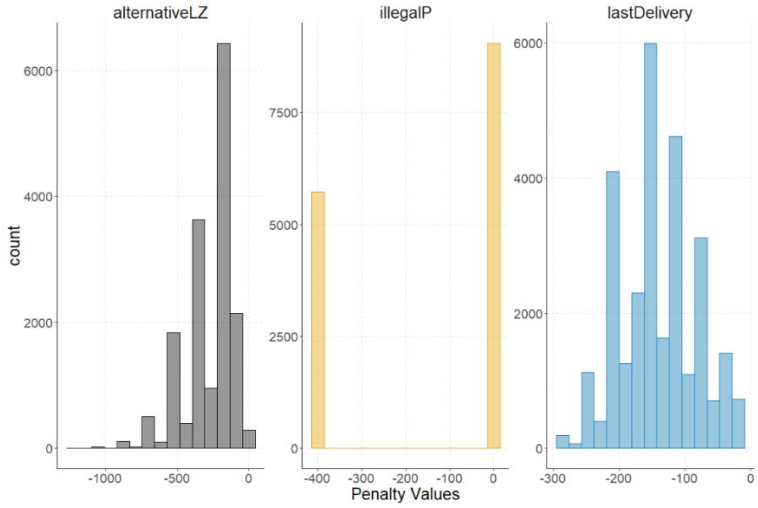


Figure 13. Penalty distribution for 1000 simulations

5.2.1 Effect Of Different Enforcement Levels

Different compliance levels of law enforcement can impact courier operation in several ways. For instance, strict enforcement can hinder delivery efficiency and disrupt operations since DVs often need to make quick stops for deliveries. Similarly, the search for an available LZ can lead to longer routes that translate into increased operational costs, reduced customer satisfaction and the timeliness of services. On the contrary, lax enforcement has a direct effect on the deterioration of safety, accessibility and fairness of the traffic network. The proposed framework can be used to analyse and compare the effect of different policies in the operation of LZs. For instance, as mentioned in section 3.1.1 we evaluate low, medium and high parking enforcement levels and the effect on DVs decision making.

Figure 14 shows the effect of this analysis for 500 simulation runs for three different DVs in the system. Under low compliance levels, DVs exhibit a risk taker behaviour, most of the time obtaining significantly lower penalties than the other strategies. However, in the long term the Illegal Parking strategy is not selected after the DVs has had a considerable exploration and gathered enough information to weight the benefits/consequences of this strategy. This effect can also be attributed to the fact that the Illegal Parking strategy has significantly higher penalty values (time converted), compared to Alternative LZ and Last Delivery. Consequently, the decision-making process to converge less rapidly to the Last Delivery strategy. On the other end, a risk averse behaviour can be seen under the medium and high enforcement levels. Under this policies DVs use the Illegal Parking strategy mostly at the beginning of the delivery period, where exploration values (ε) are high and there is still not enough information and learning to determine the best strategy. This behaviour is corrected rapidly specially under a high enforcement level. As this figure shows, once the system has converged, the Illegal Parking strategy and even the alternative LZ are only evaluated sporadically with probability ε (as described in Section 3.2.2).

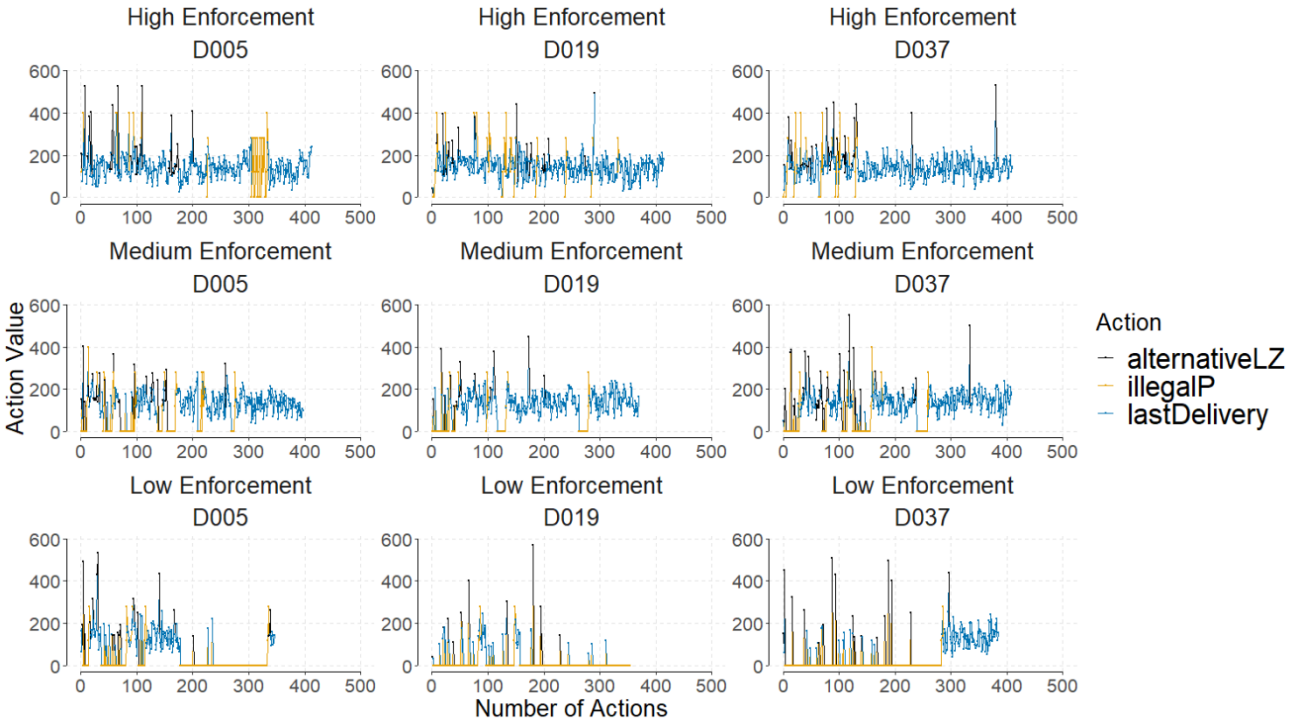


Figure 14. Action value time series for DV 005, DV019 and DV037

5.2.2 Effect of DVs decisions on traffic network performance

As stated in the introduction section, our main purpose is to find strategies that balance the courier's economic (delivery efficiency in terms of total travel time) and the network performance objectives (traffic flow variability). For this reason, to better observe the effect of DVs in terms of traffic network capacity and reliability reduction, we plot the interquartile coefficient of variation (IQRcv) of the traffic flow of the studied road network in Figure 15. The interquartile range is considered a more robust measure of spread and the median is a more robust measure of central tendency. The IQRcv can be a better option over the more popular coefficient of variation, which is based on the mean and standard deviation, have a strong connection to the normal distribution and might be too sensitive to outliers (Bonett, 2006; Doulah, 2018). We use the IQRcv as a performance evaluator of the traffic flow when DVs use the RL model (*Armed Bandit* strategy) to make parking decisions versus the use of one strategy (*Alternative LZ*, *Illegal Parking* or *Last Delivery*) for the whole simulation period. Results show that the boxplot for the *Armed Bandit* strategy exhibits a higher tail in the 4th quartile of the distribution with some outliers. Despite this, the *Armed Bandit* strategy shows a lower median than the *Alternative LZ* and *Last Delivery* strategies. The fact that the IQRcv for the *Armed Bandit* strategy is not better than *Illegal Parking* is somehow expected since the locations of the LZs in the study network were equally spread over the edges, with two illegal parking areas. This aspect greatly diminishes the spread of congestion to adjacent links and therefore the impact over the IQRcv. A similar situation occurs when DVs parked illegally are blocking access to buildings, side streets or public transport.

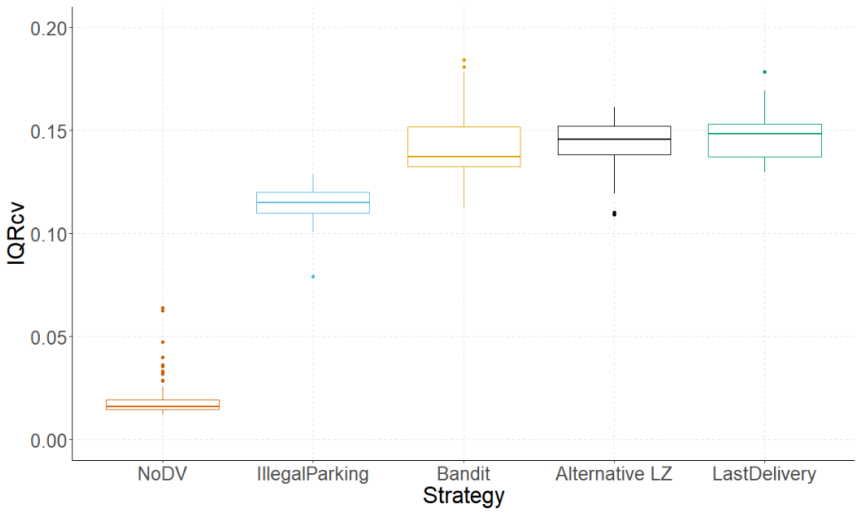


Figure 15. IQRcv distribution for *Illegal Parking*, *Alternative LZ*, *Last Delivery* and *Armed Bandit* strategies

To evaluate the results of the strategies from the couriers' perspective, we examine the additional cruising percentage (ACP) for DVs shown in Figure 16. The y-axis shows the percentage of additional cruising time to finish all delivery tasks compared to the ideal solution found with medium traffic congestion and no LZ competition. Specifically, the ideal solution is obtained by considering the minimum first- and second-level trips found with the evolutionary algorithm using a constant walking speed and average driving speed at medium congestion levels, for one single DV in the network. A zero ACP implies finishing all delivery tasks in the ideal timeframe. By using the *Armed Bandit* strategy a DV could expect around 75% chance of having ACPs lower than 60%, compared to *Alternative LZ* and *Last Delivery* strategies that have around 75% chance of ACPs lower than 100%. Similarly to the IQRcv, the *Armed Bandit* strategy does not have results as good as the *Illegal Parking strategy* which has a 75% chance of having less than 47% ACPs. This is because the *Illegal Parking* strategy offers the same benefit as parking at the desired LZ, with no penalty imposed on driving or walking time, albeit at the risk of potential fines. In summary, these results clearly demonstrate the advantages of employing RL to model DVs' decisions when encountering LZ unavailability issues in heavily congested networks. Their behavior closely resembles that of illegal parking without resorting to unlawful actions as the sole recourse.

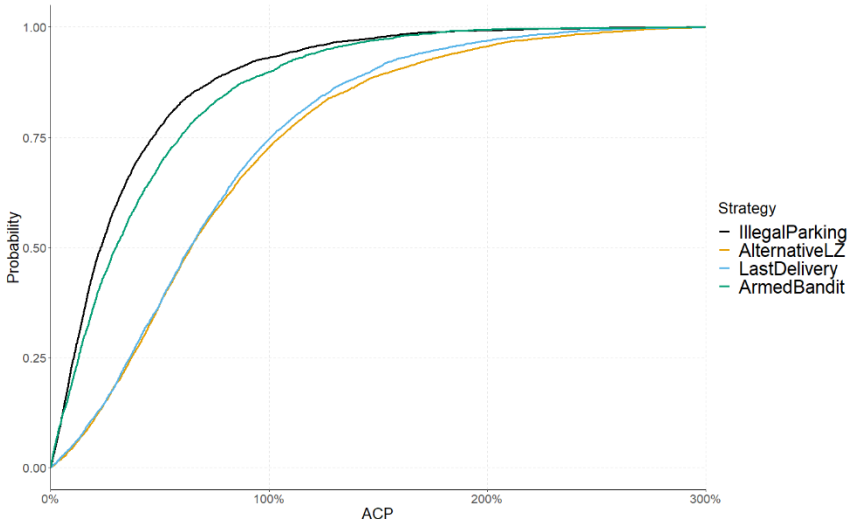


Figure 16. Results for the ACP for *Illegal Parking*, *Alternative LZ*, *Last Delivery*, and k-*Armed Bandit* strategies

6. Conclusions

By nature, the urban logistics environment is highly uncertain and difficult to foresee when planning route operations for delivery vehicles. Although the use of advanced software has given courier companies unprecedented capabilities to solve problems, there is still a large number of decisions that are left in the hands of the driver and that are causing a significant impact on the efficiency of courier companies, their drivers and their crew. In this study, we combined a simulation and optimisation framework with RL methods, and use them as a decision support tool for making parking decisions in the last-mile delivery. Despite that evidence shows that these decisions are a recurrent problem for DVs, the literature is still scarce. In fact, most of the applications of RL to logistic problems faced concerns at tactical levels, focusing on computational improvements by solving VRP variants; and at strategic levels, assessing the economic and environmental viability of urban distribution centres and/or urban consolidation centres.

Contrary to most formulations in the literature, in this study, the penalty functions are calculated based on time increases that measure the consequence of decisions compared to the initial solution. To simulate realistic behaviour in the *Illegal Parking* strategy, we defined an exponential function for generating higher ticket fine probabilities as the illegal parking time increases. Even though this approach might work when considering the risk-averse behaviour of drivers, recent studies found that courier companies are willing to bear this as a "cost of doing business" in CBD areas. This risk-taker behaviour is based on the drivers' experience that looking for a parking spot (unoccupied loading zone) can be an incredibly time-consuming task that will add more time to the delivery route than parking in one spot to make as many deliveries as possible. The short-term consequences of this strategy are high costs in parking fines, towing away fees and exhausted crews that derive in long-term consequences such as the loss of revenue and high personnel turnover rates. This approach is a desperate alternative and the best example of the lack of decision-support systems in last-mile delivery.

Since the configuration of the *Illegal Parking* strategy is not constrained by the consequences of a ticket fine when a DV chooses this option as a stand-alone solution, we can use its results as a lower bound for the ACP and the IQRcv. Results show that using the *Armed Bandit* strategy for taking en-route decisions provides the greatest advantage in terms of the minimisation of the ACPs and the variability of traffic flow, compared to *Alternative LZ* and *Last Delivery* strategies. In fact, the *Armed Bandit* strategy has ACPs that closely match the *Illegal Parking* strategy, even though is only used 29% of the time by DVs.

Albeit the model developed exhibits interesting results, it is important to highlight some limitations. First, the value of the action value matrix is updated after a DV chooses an action, independently of the LZ where is located. A more detailed approach could use Q-learning to create independent state action tables that save the penalties obtained from each LZ. Although realistic, this will imply that each LZ is a different state in the system and that every action in that state takes the vehicle to a different future one. This is possible for the *Alternative LZ* strategy, which chooses a closer LZ as a substitute, to conduct a reoptimisation process for the rest of the delivery route. Nevertheless, for the *Last Delivery* and *Illegal Parking* strategies, this concept becomes unpractical since there is not a valid reason to change the rest of the route once a DV has used these alternatives. Despite this limitation, the application of more advanced models using Q-learning or Monte Carlo methods is a promising future research application. Similarly, dynamic programming is an alternative for the solution of MDPs, this modellling approach could be incroporated within the proposed simulation-optimisation model. Some works in the literature already explore this avenue in the study of urban logistics. This direction result into additional complexities in the proposed model. The course of dimensionality of dynamic programming appears given the large action space resulting from multiple independent DVs in the system and their possible actions (with several LZs and actions to perfoms).

References

Accorsi, L., & Vigo, D. (2020). A hybrid metaheuristic for single truck and trailer routing problems. *Transportation Science*, 54(5). pp. 1351-1371.

Ahuja, R. K., & Orlin, J. B. (1997). Commentary—Developing Fitter Genetic Algorithms. *INFORMS Journal on Computing*, 9(3). pp. 251-253.

Aiura, N., & Taniguchi, E. (2005). Planning on-street loading-unloading spaces considering the behaviour of pickup-delivery vehicles. *Journal of the Eastern Asia Society for Transportation Studies*, *6*, pp. 2963-2974. DOI: https://doi.org/10.11175/easts.6.2963.

Alho, A., e Silva, J. D. A., & de Sousa, J. P. (2014). A state-of-the-art modeling framework to improve congestion by changing the configuration/enforcement of urban logistics loading/unloading bays. *Procedia-Social and Behavioral Sciences*, *111*, pp. 360-369. DOI: https://doi.org/10.1016/j.sbspro.2014.01.069

Aljohani, K., & Thompson, R. G. (2018). Optimizing the Establishment of a Central City Transshipment Facility to Ameliorate Last-Mile Delivery: a Case Study in Melbourne CBD. *City Logistics 3: Towards Sustainable and Liveable Cities*, pp. 23-46.

Amazon Last-mile Routing Research Challenge (2021). https://routingchallenge.mit.edu/

Araghi, S., Khosravi, A., Johnstone, M., & Creighton, D. (2013). A novel modular Q-learning architecture to improve performance under incomplete learning in a grid soccer game. *Engineering Applications of Artificial Intelligence*, *26*(9), pp. 2164-2171.

Aragon-Gómez, R., & Clempner, J. B. (2020). Traffic-signal control reinforcement learning approach for continuous-time markov games. *Engineering Applications of Artificial Intelligence*, *89*. DOI: https://doi.org/10.1016/j.engappai.2019.103415

Baker, L 2019, 'New York City charges UPS and FedEx millions in parking fines', Freight Waves. Retrieved: August 7/2021, available at https://www.freightwaves.com/news/ups-hit-with-22m-in-nyc-parking-fines.

Basso, R., Kulcsár, B., Sanchez-Diaz, I., & Qu, X. (2022). Dynamic stochastic electric vehicle routing with safe reinforcement learning. *Transportation Research Part E: Logistics and Transportation Review*, *157*. DOI: https://doi.org/10.1016/j.tre.2021.102496

Bektas, T, Crainic, TG & Van Woensel, V (2017). From Managing Urban Freight to Smart City Logistics Networks, in Gakis, K & Pardalos, P (eds.), *Network Design and Optimization for Smart Cities*, World Scientific, pp. 143-188. DOI: https://doi.org/10.1142/9789813200012_0007

Belenguer, J. M., Benavent, E., Martínez, A., Prins, C., Prodhon, C., & Villegas, J. G. (2016). A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. *Transportation Science*, *50*(2), pp. 735-749. DOI: https://doi.org/10.1287/trsc.2014.0571

Bono, G., Dibangoye, J. S., Matignon, L., Pereyron, F., & Simonin, O. (2018). SULFR: Simulation of Urban Logistic For Reinforcement. In *PGMRL 2018Workshop on Prediction and Generative Modeling in Reinforcement Learning*. pp. 1-5. Available at https://hal.inria.fr/hal-01847773

Bouhamed, O., Ghazzai, H., Besbes, H., & Massoud, Y. (2019). Q-learning based routing scheduling for a multi-task autonomous agent. In 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS). pp. 634-637.

Bouneffouf, D., Rish, I., & Aggarwal, C. (2020, July). Survey on applications of multi-armed and contextual bandits. In 2020 *IEEE Congress on Evolutionary Computation (CEC*) (pp. 1-8). IEEE.

Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum*, *43*(1), pp. 1-58. DOI: https://doi.org/10.1007/s00291-020-00607-8

Business Resarch Insight (2023). Last Mile Delivery Transportation Market Size, Share, Growth, and Industry Analysis by Type (Business-to-Business (B2B), Business-to-Consumer (B2C), and Customer-to-Customer (C2C)) By Application (Motorcycle, Commercial Vehicles, and Drones), Regional Insights and Forecast to 2031. Retrieved: February 19/2023. Available at: https://www.businessresearchinsights.com/market-reports/last-mile-delivery-transportation-market-101836

Campagna, A., Stathacopoulos, A., Persia, L., & Xenou, E. (2017). Data collection framework for understanding UFT within city logistics solutions. *Transportation Research Procedia*, *24*, pp. 354-361. DOI: https://doi.org/10.1016/j.trpro.2017.05.100

Casey, N., Rao, D., Mantilla, J., Pelosi, S., & Thompson, R. G. (2014). Understanding last kilometre freight delivery in Melbourne's Central Business District. *Procedia-Social and Behavioral Sciences*, *125*, pp. 326-333. DOI: https://doi.org/10.1016/j.sbspro.2014.01.1477

Cavagnini, R., Schneider, M., & Theiß, A. (2023). A granular iterated local search for the asymmetric single truck and trailer routing problem with satellite depots at DHL Group. *Networks*.

Cesa-Bianchi, N., Gentile, C., Lugosi, G., & Neu, G. (2017). Boltzmann exploration done right. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6287–6296.

Chen, Y., Qian, Y., Yao, Y., Wu, Z., Li, R., Zhou, Y., H. Hu & Xu, Y. (2019). Can sophisticated dispatching strategy acquired by reinforcement learning?-a case study in dynamic courier dispatching system. *arXiv preprint arXiv:1903.02716*.

Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), pp. 1086-1095. DOI: https://doi.org/10.1109/TITS.2019.2901791

Comi, A., Schiraldi, M. M., & Buttarazzi, B. (2018). Smart urban freight transport: tools for planning and optimising delivery operations. *Simulation Modelling Practice and Theory*, *88*, 48-61. DOI: https://doi.org/10.1016/j.simpat.2018.08.006.

Cuda, R., Guastaroba, G., & Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, 55, 185-199. DOI: https://doi.org/10.1016/j.cor.2014.06.008

Crainic, T. G., Ricciardi, N., & Storchi, G. (2004). Advanced freight transportation systems for congested urban areas. *Transportation Research Part C: Emerging Technologies*, *12*(2), pp. 119-137. DOI: https://doi.org/10.1016/j.trc.2004.07.002

Dablanc, L. (2011). City distribution, a key element of the urban economy: guidelines for practitioners. In C. Macharis & S. Melo (Eds.), City distribution and Urban freight transport: Multiple perspectives (p. 261). *NECTAR Series on Transportation and Communications Networks Research*.

Dablanc, L., & Beziat, A. (2015). Parking for freight vehicles in dense urban centers-The issue of delivery areas in Paris. *Marne la Vallee, France*. Available at: https://www.metrans.org/assets/research/MF14-3%202d_Parking%20for%20Freight%20Vehicles%20Final%20Report_070815_0.pdf

Dalla Chiara, G., & Cheah, L. (2017). Data stories from urban loading bays. *European Transport Research Review*, *9*(4), 1-16. DOI: https://doi.org/10.1007/s12544-017-0267-3

Dalla Chiara, G., & Goodchild, A. (2020). Do commercial vehicles cruise for parking? Empirical evidence from Seattle. *Transport Policy*, *97*, pp. 26-36. DOI: https://doi.org/10.1016/j.tranpol.2020.06.013

Dalla Chiara, G., Cheah, L., Azevedo, C. L., & Ben-Akiva, M. E. (2020). A policy-sensitive model of parking choice for commercial vehicles in urban areas. *Transportation Science*, *54*(3), 606-630.

Delaître, L., & Routhier, J. L. (2010). Mixing two French tools for delivery areas scheme decision making. *Procedia-Social and Behavioral Sciences*, *2*(3), pp. 6274-6285. DOI: https://doi.org/10.1016/j.sbspro.2010.04.037

Delaitre, L. (2009). A new approach to diagnose urban delivery areas plans. In *2009 International Conference on Computers & Industrial Engineering*. pp. 991-998. DOI: https://doi.org/10.1109/ICCIE.2009.5223953

Dezi, G., Dondi, G., & Sangiorgi, C. (2010). Urban freight transport in Bologna: Planning commercial vehicle loading/unloading zones. *Procedia-Social and Behavioral Sciences*, *2*(3), pp. 5990-6001. DOI: https://doi.org/10.1016/j.sbspro.2010.04.013

Ewedairo, K., Chhetri, P., & Jie, F. (2018). Estimating transportation network impedance to last-mile delivery: A Case Study of Maribyrnong City in Melbourne. *The International Journal of Logistics Management*. 29(1), pp. 110–130. DOI: https://doi.org/10.1108/IJLM-10-2016-0247

Firdausiyah, N., Taniguchi, E., & Qureshi, A. G. (2019). Modeling city logistics using adaptive dynamic programming based multi-agent simulation. *Transportation Research Part E: Logistics and Transportation Review*, *125*, pp. 74-96. DOI: https://doi.org/10.1016/j.tre.2019.02.011

Guillet, M., Hiermann, G., Kröller, A., & Schiffer, M. (2022). Electric Vehicle Charging Station Search in Stochastic Environments. *Transportation Science*, *56*(2), pp. 483-500.

Guillet, M., & Schiffer, M. (2022). Coordinated Charging Station Search in Stochastic Environments: A Multi-Agent Approach. *arXiv preprint arXiv:2204.14219*.

Guo, X., Ren, Z., Wu, Z., Lai, J., Zeng, D., & Xie, S. (2020, November). A deep reinforcement learning based approach for AGVs path planning. In 2020 Chinese automation congress (CAC). pp. 6833-6838.

Gupta, A., Ghosh, S., & Dhara, A. (2022). Deep Reinforcement Learning Algorithm for Fast Solutions to Vehicle Routing Problem with Time-Windows. In *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*. pp. 236-240. DOI: https://doi.org/10.1145/3493700.3493723

Han, L. D., Chin, S. M., Franzese, O., & Hwang, H. (2005). Estimating the impact of pickup-and delivery-related illegal parking activities on traffic. *Transportation Research Record*, *1906*(1), pp. 49-55. DOI: https://doi.org/10.1177/0361198105190600106.

Hildebrandt, F. D., Thomas, B. W., & Ulmer, M. W. (2023). Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research*, 150, 106071.

Holguín-Veras, J., Leal, J. A., Sánchez-Diaz, I., Browne, M., & Wojtowicz, J. (2020). State of the art and practice of urban freight management: Part I: Infrastructure, vehicle-related, and traffic operations. *Transportation Research Part A: Policy and Practice*, *137*, pp. 360-382.

Iwan, S., Kijewska, K., Johansen, B. G., Eidhammer, O., Małecki, K., Konicki, W., & Thompson, R. G. (2018). Analysis of the environmental impacts of unloading bays based on cellular automata simulation. *Transportation Research Part D: Transport and Environment*, *61*, pp. 104-117. DOI: https://doi.org/10.1016/j.trd.2017.03.020.

Jaller, M., Holguín-Veras, J., & Hodge, S. D. (2013). Parking in the city: Challenges for freight traffic. *Transportation research record*, *2379*(1), pp. 46-56. DOI: https://doi.org/10.3141/2379-06.

Jahanshahi, H., Bozanta, A., Cevik, M., Kavuk, E. M., Tosun, A., Sonuc, S. B., B. Kosuku & Başar, A. (2021). A Deep Reinforcement Learning Approach for the Meal Delivery Problem. *arXiv preprint* http://arxiv.org/abs/2104.12000

James, J. Q., Yu, W., & Gu, J. (2019). Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(10), pp. 3806-3817.

Lamas-Fernandez, C., Martinez-Sykora, A., McLeod, F., Bektaş, T., Cherrett, T., & Allen, J. (2023). Improving last-mile parcel delivery through shared consolidation and portering: A case study in London. *Journal of the Operational Research Society*. pp. 1-12.

Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Vol. 1. Cambridge University Press.

Letnik, T., Farina, A., Mencinger, M., Lupi, M., & Božičnik, S. (2018). Dynamic management of loading bays for energy efficient urban freight deliveries. *Energy*, *159*, pp. 916-928. DOI: https://doi.org/10.1016/j.energy.2018.06.125.

Long, G, 2000, 'Acceleration Characteristics of Starting Vehicles', Transportation Research Record, vol. 1737, no. 1, pp. 58–70. DOI: https://doi.org/10.3141/1737-08.

Luke, S. (2013). Essentials of Metaheuristics: a set of undergraduate lecture notes, Vol. 2, Lulu.

Malik, L., Sánchez-díaz, I., Tiwari, G., & Woxenius, J. (2017). Urban freight-parking practices: The cases of Gothenburg (Sweden) and Delhi (India). *Research in Transportation Business & Management*, 24(October 2016). pp. 37–48. DOI: https://doi.org/10.1016/j.rtbm.2017.05.002

Marcia, S. (2009). Improving Freight Movement in Delaware Central Business. In Institute for Public Administration, College of Education & Public Policy.

Martinez-Sykora, A, McLeod, F, Lamas-Fernandez, C, Bektas, T, Cherrett, T & Allen, J (2020), 'Optimised solutions to the last-mile delivery problem in London using a combination of walking and driving', Annals of Operations Research, vol. 295, pp. 645–693. DOI: https://doi.org/10.1007/s10479-020-03781-8

McKinsey & Co. (2019). The future of parcel delivery: Drones and disruption The Next Normal. Retrieved: November 29/2021. Available at: https://www.mckinsey.com/featured-insights/the-next-normal/parcel-delivery

McLeod, F., & Cherrett, T. (2011). Loading bay booking and control for urban freight. *International Journal of Logistics Research and Applications*, *14*(6). pp. 385-397. DOI: https://doi.org/10.1080/13675567.2011.641525.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, *7*(4). pp. 326-329.

Munuzuri, J., Racero, J., & Larrañeta, J. (2002). Parking search modelling in freight transport and private traffic simulation. *WIT Transactions on The Built Environment*, *60*. pp. 335-344. DOI: https://doi.org/10.2495/UT020331

Muriel, J. E., Zhang, L., Fransoo, J. C., & Perez-Franco, R. (2022). Assessing the impacts of last mile delivery strategies on delivery vehicles and traffic network performance. *Transportation Research Part C: Emerging Technologies*, *144*, 103915.

Naeem, M., Rizvi, S. T. H., & Coronato, A. (2020). A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access*, *8*, 209320-209344.

Nagel, K., & Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de physique I*, *2*(12). pp. 2221-2229. DOI: https://doi.org/10.1051/jp1:1992277

Nourinejad, M., Wenneman, A., Habib, K. N., & Roorda, M. J. (2014). Truck parking in urban areas: Application of choice modelling within traffic microsimulation. *Transportation Research Part A: Policy and Practice*, *64*. pp. 54-64. DOI: https://doi.org/10.1016/j.tra.2014.03.006

Nowé, A., & Brys, T. (2016). A gentle introduction to reinforcement learning. In *International Conference on Scalable Uncertainty Management* (pp. 18-32). September 2016, Springer, Cham.

Pinto, R., Golini, R., & Lagorio, A. (2016). Loading/unloading lay-by areas location and sizing: a mixed analytic-Monte Carlo simulation approach. *IFAC-PapersOnLine*, *49*(12). pp. 961-966. DOI: https://doi.org/10.1016/j.ifacol.2016.07.900

Potvin, J. Y. (2009). State-of-the art review—Evolutionary algorithms for vehicle routing. *INFORMS Journal on computing*, 21(4). Pp. 518-548.

Qiang, W., & Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. In *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)* Budapest, Hungary, 3–7 July 2011 (pp. 1143-1146). DOI: https://doi.org/10.1109/MEC.2011.6025669

Qin, W., Sun, Y. N., Zhuang, Z. L., Lu, Z. Y., & Zhou, Y. M. (2021). Multi-agent reinforcement learning-based dynamic task assignment for vehicles in urban transportation system. *International Journal of Production Economics*, *240*, 108251. DOI: https://doi.org/10.1016/j.ijpe.2021.108251

Rawat, K., Katiyar, V. K., & Gupta, P. (2012). Two-lane traffic flow simulation model via cellular automaton. *International Journal of vehicular technology*, *2012*. DOI: https://doi.org/10.1155/2012/130398

Reed, S., Campbell, A. M., & Thomas, B. W. (2022). The Value of Autonomous Vehicles for Last-mile Deliveries in Urban Environments. *Management Science*. DOI: https://doi.org/10.1287/mnsc.2020.3917.

Roca-Riu, M., Fernández, E., & Estrada, M. (2015). Parking slot assignment for urban distribution: Models and formulations. *Omega*, *57*. pp. 157-175. DOI: https://doi.org/10.1016/j.omega.2015.04.010

Roca-Riu, M., Cao, J., Dakic, I., & Menendez, M. (2017). Designing dynamic delivery parking spots in urban areas to reduce traffic disruptions. *Journal of Advanced Transportation*, *2017*. DOI: https://doi.org/10.1155/2017/6296720.

Rolf, B., Jackson, I., Müller, M., Lang, S., Reggelin, T., & Ivanov, D. (2023). A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research*, 61(20), 7151-7179. https://doi.org/10.1080/00207543.2022.2140221

Saravanan, M., & Ganeshkumar, P. (2020). Routing using reinforcement learning in vehicular ad hoc networks. *Computational Intelligence*, *36*(2). pp. 682-697. DOI: https://doi.org/10.1111/coin.12261

Šemrov, D., Marsetič, R., Žura, M., Todorovski, L., & Srdic, A. (2016). Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research Part B: Methodological*, *86*. pp. 250-267. DOI: https://doi.org/10.1016/j.trb.2016.01.004

Shiftan, Y., & Burd-Eden, R. (2001). Modeling response to parking policy. *Transportation Research Record*, *1765*(1). pp. 27-34. DOI: https://doi.org/10.3141/1765-05

Shoup, D. C. (2021). *The high cost of free parking*. Routledge.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., & Van Woensel, T. (2023). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, 304(3). pp. 865-886. DOI: https://doi.org/10.1016/j.ejor.2022.02.022

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Second Edition, vol. 2. MIT press.

Tamayo, S., Gaudron, A., & de La Fortelle, A. (2017). Loading/unloading spaces location and evaluation: an approach through real data. In *10th International Conference on City Logistics*. pp. 161–180. Wiley, DOI: https://doi.org/10.1002/9781119425472.ch9

Teo, J. S. E., Taniguchi, E., & Qureshi, A. G. (2015). Evaluation of urban distribution centers using multiagent modeling with geographic information systems. *Transportation Research Record*, vol *2478*(1). pp. 35-47. DOI: https://doi.org/10.3141/2478-05

Teo, J. S., Taniguchi, E., & Qureshi, A. G. (2012). Evaluating city logistics measure in e-commerce with multiagent systems. *Procedia-Social and Behavioral Sciences*, *39*. pp. 349-359.

Thompson, R. G., & Zhang, L. (2018). Optimising courier routes in central city areas. *Transportation Research Part C: Emerging Technologies*, *93*. pp. 1-12. DOI: https://doi.org/10.1016/j.trc.2018.05.016

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1). pp. 1-21.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., & Velasco, N. (2010). GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, *23*(5). pp. 780-794. DOI: https://doi.org/10.1016/j.engappai.2010.01.013.

Wang, X., Ke, L., Qiao, Z., & Chai, X. (2020). Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE transactions on cybernetics*, *51*(1). pp. 174-187. DOI: https://doi.org/10.1109/TCYB.2020.3015811

Wangapisit, O., Taniguchi, E., Teo, J. S., & Qureshi, A. G. (2014). Multi-agent systems modelling for evaluating joint delivery systems. *Procedia-Social and Behavioral Sciences*, *125*. pp. 472-483. DOI: https://doi.org/10.1016/j.sbspro.2014.01.1489

Wei, H., Zheng, G., Gayah, V., & Li, Z. (2019). A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117*.

Weinberger, R. R., Millard-Ball, A., & Hampshire, R. C. (2020). Parking search caused congestion: Where's all the fuss?. *Transportation Research Part C: Emerging Technologies*, 120, 102781.

Yan, Y., Chow, A. H., Ho, C. P., Kuo, Y. H., Wu, Q., & Ying, C. (2022). Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities. *Transportation Research Part E: Logistics and Transportation Review*, 162, 102712. DOI: https://doi.org/10.1016/j.tre.2022.102712

Yang, K., Roca-Riu, M., & Menéndez, M. (2019). An auction-based approach for prebooked urban logistics facilities. *Omega*, *89*. pp. 193-211. DOI: https://doi.org/10.1016/j.omega.2018.10.005

Ye, Q., Feng, Y., Candela, E., Escribano Macias, J., Stettler, M., & Angeloudis, P. (2022). Spatial-Temporal Flows-Adaptive Street Layout Control Using Reinforcement Learning. *Sustainability*, *14*(1), 107. DOI: https://doi.org/10.3390/su14010107

Yu, J. J. Q., Yu, W., & Gu, J. (2019). Online Vehicle Routing with Neural Combinatorial Optimization and Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(10). pp. 3806–3817. DOI: https://doi.org/10.1109/TITS.2019.2909109

Yu, L., Zhang, C., Jiang, J., Yang, H., & Shang, H. (2021). Reinforcement learning approach for resource allocation in humanitarian logistics. *Expert Systems with Applications*, vol *173*. DOI: https://doi.org/10.1016/j.eswa.2021.114663

Zhang, L., & Thompson, R. G. (2019). Understanding the benefits and limitations of occupancy information systems for couriers. *Transportation Research Part C*, 105. pp. 520–535. DOI: https://doi.org/10.1016/j.trc.2019.06.013

Zhao, J., Mao, M., Zhao, X., & Zou, J. (2020). A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*, 22(11). pp. 7208-7218.

Zheng, N., & Geroliminis, N. (2016). Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing. Transportation Research Part B: Methodological, 83. pp. 36-58. DOI: https://doi.org/10.1016/j.trb.2015.10.008.

Appendix A

The mathematical formulation for the delivery route planning problem is given below. problem formulation:

Sets:

B: $set\ of\ LZs$

C: $set\ of\ Customers$

V_1: $B \cup \{0\}\ set\ of\ vertices\ that\ can\ be\ visited\ in\ the\ first\ level\ trip; 0\ is\ the\ depot$

V_2: $C \cup B\ set\ of\ vertices\ that\ can\ be\ visited\ in\ the\ second\ level\ trip$

V^j: $C \cup \{j\}\ set\ of\ vertices\ that\ can\ be\ visited\ in\ the\ second\ level\ trip\ rooted\ at\ LZ\ j$

Parameters:

α: $weighting\ parameter\ of\ relative\ importance\ of\ driving\ vs\ walking$

$c_{ij}(i,j\ \in V_1)$: $distance\ from\ LZ\ i\ to\ j\ in\ the\ first\ level$

$d_{lm}(l,m\ \in V_2)$: $distance\ from\ customer\ l\ to\ m\ in\ the\ second\ level$

Variables:

$y_{ij} = 1\ if\ and\ only\ if\ the\ truck\ traverse\ the\ edge\ (i,j)(i,j\ \in V_1)$

$x_{lm}^j = 1\ if\ and\ only\ if\ edge\ (l,m)(l,m \in V^j)\ is\ traversed\ on\ foot$

Objective function:

$$\min \alpha \sum_{i\in V_1}\sum_{j\in V_1} c_{ij}\, y_{ij} + (1-\alpha)\sum_{j\in B}\sum_{l\in V_2}\sum_{m\in V_2} d_{lm} x_{lm}^j \tag{1}$$

Constraints:

$$\sum_{i\in V_1} y_{ij} \le 1, \forall j \in B \tag{2}$$

$$\sum_{i\in V_1} y_{ji} \le 1, \forall j \in B \tag{3}$$

$$\sum_{i\in V_1} y_{ij} = \sum_{k\in V_1} y_{jk}\ , \forall j \in B \tag{4}$$

$$\sum_{j\in B} y_{j0} = 1, \tag{5}$$

$$\sum_{j\in B} y_{0j} = 1, \tag{6}$$

$$\sum_{i\in V'}\sum_{j\in V'} y_{ij} \le |V'| - 1\ , \forall V' \subseteq B;\ |V'| \ge 2 \tag{7}$$

$$x_{lm}^j \le \sum_{i\in V_1} y_{ij}\ , \forall l,m \in V_2, \forall j \in B \tag{8}$$

$$\sum_{l\in V_2}\sum_{j\in B} x_{lm}^j = 1, \forall m \in C \tag{9}$$

$$\sum_{l\in V_2} x_{lm}^j = \sum_{o\in V_2} x_{mo}^{,j}\ , \forall m \in C, \forall j \in B \tag{10}$$

$$\sum_{l\in C} x_{jl}^j = \sum_{o\in C} x_{oj}^j\ , \forall j \in B \tag{11}$$

$$\sum_{l \in V'} \sum_{m \in V'} x_{lm}^{j} \leq |V'| - \gamma(V'), \forall j \in B, \forall V' \subseteq B, |V'| \geq 2 \quad (12)$$

$$y_{ij} \in \{0,1\}, \forall i, j \in V_1, i \neq j \quad (13)$$

$$x_{lm}^{j} \in \{0,1\}, \forall j \in B, l, m \in V^{j}, l \neq m \quad (14)$$

The objective function (1) minimises the total weighted delivery distance, which is composed of the driving and walking components. The value of the weighting parameter α directly affects the routing and parking behaviour. Higher values will likely result in shorter first-level trips and longer second-level trips. By adjusting α, we can give more relevance to the driving or walking related cost in a single objective function. Since driving incurs labour, environmental and vehicle costs whilst walking only incurs labour costs, tuning α can also give different importance to the vehicle (and environmental) or labour cost .Constraints (2) and (3) guarantee that each LZ is visited by one DV at most once in the first level. Constraint (4) is a flow conservation constraint for the first-level trips. Constraints (5) and (6) guarantee that a first-level trip starts and ends at the initial entry point. Constraint (7) is subtour elimination constraints for the first-level trips. Constraint (8) guarantees that the second-level trip starts at a LZ visited by the vehicle in the first-level trip. Constraint (9) states that each customer is visited only once. Constraint (10) guarantees connectivity in the second-level trips. Constraint (11) guarantees that the second-level trip starts and ends at the vehicle. Constraint (12) avoids sub-tours in the second-level trips, where $\gamma(V')$ is the minimum number of second-level trips needed to serve the demand of the customers $V' \subseteq C$. Constraints (13) and (14) are binary variables constraints.