

Joint Optimisation of Drone Routing and Battery Wear for Sustainable Supply Chain Development: A Mixed-integer Programming Model Based on Blockchain-enabled Fleet Sharing

Yang Xia,^{1,2} Wenjia Zeng,^{1,2} Xinjie Xing,³ Yuanzhu Zhan,³ Kim Hua Tan,⁴ Ajay Kumar⁵

¹ *Research Center for Modern Logistics, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China*

² *Department of Industrial Engineering, Tsinghua University, Beijing 100084, China*

³ *Management School, University of Liverpool, UK*

⁴ *Nottingham University Business School, University of Nottingham, UK*

⁵ *AIM Research Centre on Artificial Intelligence in Value Creation, EMLYON Business School, France*

Abstract

Alongside the rise of ‘last-mile’ delivery in contemporary urban logistics, drones have demonstrate commercial potential, given their outstanding triple-bottom-line performance. However, as a lithium-ion battery-powered device, drones’ social and environmental merits can be overturned by battery recycling and disposal. To maintain economic performance, yet minimise environmental negatives, fleet sharing is widely applied in the transportation field, with the aim of creating synergies within industry and increasing overall fleet use. However, if a sharing platform’s transparency is doubted, the sharing ability of the platform will be discounted. Known for its transparent and secure merits, blockchain technology provides new opportunities to improve existing sharing solutions. In particular, the decentralised structure and data encryption algorithm offered by blockchain allow every participant equal access to shared resources without undermining security issues. Therefore, this study explores the implementation of a blockchain-enabled fleet sharing solution to optimise drone operations, with consideration of battery wear and disposal effects. Unlike classical vehicle routing with fleet sharing problems, this research is more challenging, with multiple objectives (i.e., shortest path and fewest charging times), and considers different levels of sharing abilities. In this study, we propose a mixed-integer programming model to formulate the intended problem and

solve the problem with a tailored branch-and-price algorithm. Through extensive experiments, the computational performance of our proposed solution is first articulated, and then the effectiveness of using blockchain to improve overall optimisation is reflected, and a series of critical influential factors with managerial significance are demonstrated.

Keywords: Drone-assisted delivery, sustainable supply chain management, blockchain, mixed-integer programming model

1. Introduction

Alongside the rise of ‘last-mile’ delivery in contemporary urban logistics, drone delivery has rapidly developed and provides ever-increasing applications, given its outstanding triple-bottom-line performance. Commercially, drone delivery helps reduce labour costs with its pilot-free feature, enables faster travel speed than traditional truck-load delivery, and provides a road network constraint-free transport solution (Agatz et al., 2018). Socially, drone delivery can ease traffic congestion (Dukkanci et al., 2021) and can be used to support post-disaster resilience by providing risk assessment, mapping and temporary communication network creation (Chowdhury et al., 2017; Tang and Veelenturf, 2019). Most importantly, in contrast to conventional means of transport, drone delivery can significantly contribute to companies’ green agenda by reducing greenhouse gas emissions (Goodchild and Toy, 2018). Driven by the merits of drone delivery, an increasing number of drone applications are seen in various business fields. For instance, German logistics giant DHL successfully flew its Parcelcopter 4.0 over Lake Victoria with a 60 km flight in 2018 (DHL, 2018); Amazon has implemented its Prime Air system by deploying drones to service its customers since 2016 (Amazon, 2020); and Chinese e-commerce giant, JD.com, has already established 150 operational sites for delivery drones since 2017 (Unmanned Cargo, 2017).

However, as a lithium-ion battery-powered device, the use of drones is no different from other electric devices, and faces battery disposal problems, meaning

that all the sustainable goodwill offered by drone delivery could be largely offset. As reported by Forbes (Rapier, 2020), the exploding sales of electric devices, especially vehicles, will result in 11 million metric tons of used lithium-ion batteries reaching the end of their service lives in 2030, while, unfortunately, the recycle rate for lithium-ion batteries globally is below 5%. When such an enormous quantity of batteries is disposed at landfills, subsequent environmental problems—such as air contamination and water supply and soil pollution—will be inevitable and severe (Katwala, 2018). As stated by Edge et al. (2021), lithium-ion batteries wear down with each charge and the maximal capacity is degraded gradually until it cannot be charged again. Therefore, optimising the lithium-ion battery charging plan is crucial for prolonging its lifespan, and will ultimately help mitigate battery disposal-caused environmental consequences. This explains the prevalence of relevant initiatives introduced in the electric vehicle industry (e.g., Pelletier et al., 2017; Yang and Sun, 2015).

However, because of the differences stemming from drone batteries and drone-assisted delivery, solutions applied to address battery-related operational issues in the electric vehicle industry have very limited effectiveness for drones. First, drone batteries have much smaller capacity than those of electric vehicles. In general, battery capacities of current electric vehicles range from 30 to 100 kWh, with voltage from 400 to 800 V (i.e., 37,500 to 250,000 mAh) (Pollard, 2020), whereas battery capacities for drones can range from 2,700 to 40,000 mAh (Figliozzi et al., 2018). As a result, drones' service coverage is highly constrained and there is little space for battery performance optimisation. Second, in contrast to electric vehicles, drones do not have a public charging network on which to rely; rather, every operator maintains its own recharging facility to support its operations. Therefore, existing approaches in the electric vehicle sector, such as optimising recharging network design and vehicle recharging decisions with given grid systems, are difficult to directly incorporate.

Although the above issues limit battery optimisation with an individual operator, the battery capacity and charging facility constraints could be alleviated if multiple operators agreed to share their fleets. If one operator's drones could be dispatched and

collected by other operators, the service coverage of this operator could be significantly expanded and more business opportunities could be fulfilled. The rise of blockchain solution is proposed to provide a trust-free sharing system that could automatically create an immutable, consensually agreed and publically available record of past transactions, thereby mitigating trust issues in a peer-to-peer system (Greiner and Wang, 2015; Hastig and Sodhi, 2020). A blockchain refers to a cryptographically secured distributed ledger with a decentralised consensus mechanism (Wang et al., 2017a). It securely keeps data shared among its users and allows users to transact valuable assets in a public and pseudonymous setup, without reliance on an intermediary or central authority (Glaser, 2017). As Hawlitschek et al. (2018) stated, mutual trust is a fundamental prerequisite for a sharing economy, as it determines the maximum sharing ability (i.e., the sum of shared information and resources from all participants) that such a sharing system can achieve. With the use of blockchain for sharing economy, smart contracts can be created to unite both operators and customers, proceed payments, and collect data under a trustworthy process. Most importantly, blockchain stands out as a sharing platform solution due to its superior data security and its advantages in ride sharing are thus appreciated in real world. For example, a Russian company Darenta implemented a blockchain-enabled car sharing platform that allows the information of shared cars and drivers traceable but the associated transactions are free from cyber-attacks (Darenta, 2021). In particular, this sharing platform is similar to the idea of Airbnb where customers can lease their own vehicles out or lease other's vehicles in, but the foundation for the trust of the community is not built upon reviews. Because of blockchain technology, all the on-the-road data are completely recorded and transparently shared, customers who lease their vehicles out has no worry about their assets are mis-treated by the leasees. On the leasees' side, blockchain keeps them peace of mind as they can have full information about their rented capacity although it is sourced from someone the leasee has no knowledge of. For the proposed drone sharing operation, although fleet capacities are only shared amongst operators, the purpose of adopting blockchain is the same. After the blockchain-enabled platform is offered by someone (e.g. an

independent service provider), all drone operators are the “customers” of the platform and information about each operator as well as their fleet conditions become transparent to each participants. Consequently, trust is enhanced as each operator has no need to worry about the conditions of the sourced fleet nor the safety their own fleet that is being shared with other operators. Given that drone delivery is highly constrained by load capacity, battery capacity and infrastructure resources, any subtle increase in the shared resources could make a significant difference. Hence, it makes both economical and practical sense to manage drone operations on a blockchain-enabled sharing system.

Regarding the above discussion, this study aimed to support operations managers in facilitating sustainable supply chain development by optimising the delivery route and extending drone battery lifespan through a blockchain-enabled drone sharing approach. The remainder of this paper is organised as follows. Section 2 reviews the relevant literature and identifies the research gaps, while Section 3 explores the underlying problems in more detail and formulates them mathematically. Section 4 presents the methodology used in detail, and Section 5 conducts various numerical experiments to illustrate the novelty of this research. Finally, Section 6 discusses the main research findings and highlights directions for future studies.

2. Literature review

Considering the relevance of this research, this section will review existing studies from drone routing optimisation and blockchain-enabled sharing economy to indicate the academic significance of this study.

2.1. Drone routing and operation optimisation

Research on drone delivery originally derived from investigations into vehicle routing problems (VRPs). In urban logistics, especially in the last-mile delivery field, the key objective of VRP and its variants is determining a set of optimal routes performed by vehicles with limited capacity and operational constraints to serve a certain group of customers (e.g., Lahyani et al., 2015; Laporte, 2009; Villegas et al.,

2013; Wang et al., 2017b; Wang et al., 2020a; Wang and Sheu, 2019). By incorporating drone delivery to replace or assist the traditional truck-based transport mode, new challenges are arising. For instance, drone service is much more constrained by limited capacity, limited flight range and so forth. When drones operate solely or collaboratively with trucks to serve different customers, existing VRP studies are difficult to apply, as model formulation and the associated solution strategy must be adaptive to the research context (Wang and Sheu, 2019).

In addition, when sustainability is considered for traditional VRP problems (e.g., Govindan et al., 2014; Govindan et al., 2019), the focus is minimisation of mileage-related carbon footprint, while, for drone problems, the major sustainability issue is effective battery usage (Katwala, 2018). Therefore, a new research stream has developed to incorporate features of drone delivery in classic VRPs. In particular, San et al. (2016) detailed the process of managing a fleet of heterogeneous drones to serve customers at different locations. The study showed that the key operational challenge of drone application is limited load capacity. As a result, many demands must be strategically fulfilled by multiple trips. Choi and Schonfeld (2017) investigated an automated drone delivery system by considering drone battery capacity, load capacity and flight range constraints. Compared with classic VRP studies, drone-based VRP is challenging because certain drone features significantly limit their service range. Therefore, while the above studies sought to increase the effective flying journey of drones to reduce unnecessary flying distance, another stream of research aims to adopt different initiatives to improve drones' service coverage.

To expand the service range of drones, Murray and Chu (2015) introduced the 'travelling salesman' problem with drone applications (TSP-D), whereby a drone collaborates with a truck to distribute customer parcels at a minimum completion time of two vehicles. The findings summarised two types of drone/truck tandem scheduling problems—the flying sidekick travelling salesman problem (FSTSP) and parallel drone scheduling travelling salesman problem (PDSTSP)—to improve the operation of drone-and-truck coordination. Specifically, the first refers to a drone working in tandem with a delivery truck, while the latter describes a scenario with a distribution

centre located in close proximity to customers on top of FSTSP. By applying a route-first, cluster-second heuristic approach, their solution is effective with up to 10 customers. Following a similar methodology, Ha et al. (2018) further improved this heuristic approach with a local search and greedy randomised adaptive search procedure (GRASP) technique, which allowed it to be effective with 50 and 100 customer nodes.

By modelling TSP-D with integer programming, Agatz et al. (2018) proposed both an exact solution and a heuristic to solve the problem. A worst-case approximation guarantee was given for a theoretical analysis conduct and the performance of the heuristic solution was compared with the exact one through an experimental comparison. In a similar context, Wang et al. (2017a) studied TSP-D with a focus on deriving worst-case bounds for ratios of the total delivery time of different delivery options. Marinelli et al. (2017) further extended FSTSP studies by considering whether drone launch and rendezvous operations can be performed throughout its route arcs. Ham (2018) extended the PDSTSP by using drones for both drop-off and pickup activities. Ulmer and Thomas (2018) considered a heterogeneous fleet and used an approximate dynamic programming solution to solve the TSP-D, with customer orders realised dynamically during each working shift. Yurek and Ozmutlu (2018) provided an approach for solving TSP-D based on decomposition of the problem into two components. Wang and Sheu (2019) further allowed docking hubs in a TSP-D problem, which provided flexibility for coordination between trucks and drones. Moshref-Javadi et al. (2020) modelled a joint routing problem of a truck with one or multiple drones, and designed a hybrid tabu search-simulated annealing algorithm to improve the customer waiting situation.

Further, D'Andrea (2014) stated that the energy consumption rate for drones should vary depending on the condition of the payload. This is significant for drones because a drone's average payload is much higher than its own weight. To enhance drone applications' feasibility, many existing studies have explored the TSP-D problem in different situations. In particular, given the non-linear components, some research has employed linear approximation-based techniques to reduce their

associated complexity (e.g., Dorling et al., 2017; Jeong et al., 2019; Troudi et al., 2018). Further, to identify the key parameters affecting the energy consumption behaviour of drones, some studies have discussed the similarities and differences among various models (e.g., Kirschstein, 2020; Zhang et al., 2021). Moreover, some research has directly formulated drone energy consumption in a non-linear form and solved the TSP-D with considering non-linear energy consumption with exact solutions (e.g., Chen et al., 2020; Dukkanci et al., 2021). All these research streams have shifted the focus of TSP-D problems to energy and battery considerations, and have enhanced the practicability of the research by modelling energy consumption more comprehensively. However, there is a lack of research investigating how operators can effectively extend the use of drone batteries and reduce the effects of battery capacity constraints on drones' daily operations.

To fill this gap, some recent research introduced battery recharging or replacing facilities to extend the overall length of one flight of a drone. For example, Hong et al. (2018) investigated a range-restricted recharging station coverage model for drone delivery service planning. Asadi and Pinkley (2021) studied scheduling, allocation and inventory replenishment at battery swap stations to overcome battery degradation for drones and electric vehicles. Cokyasar (2021) introduced automated battery swapping machines (ABSMs) to the design of a drone-based service network, which is greatly promoted by the United States government. Although introduction of these facilities can extend the service time of a drone, some important issues have been identified and remain unaddressed. First, existing studies on battery swapping or recharging facilities mainly focused on overall network design, rather than charging optimisation for individual drones. Thus, there is a lack of decision support to derive a specific plan for drone routing and battery charging in different situations when it comes to daily operations. Second, the lifespans of lithium-ion batteries vary concerning how often and how much electricity is left when the battery is recharged. According to Edge et al. (2021), the less residual electricity in a battery before each charging, a higher capacity fade the battery will have. This battery degradation effect can greatly affect drone operations, especially regarding the fixed costs for battery

replacements and environmental effects. However, this has not been explored in existing studies. Third, similar to the essence of recharging points and ABSMs, drone fleet sharing can be an approach for extending the flight range of drones, and does not require a huge upfront investment. However, it requires an effective and trustworthy platform to connect all participating operators for collaboration. Unlike ABSMs, which only extend drone flight range yet do not alter fleet size, drone sharing can significantly increase each operator's total capacity and enable more profitable business opportunities. Fourth, considering the significant environmental effects of battery disposal, it is worth including this factor with other economic indicators for joint optimisation. It is both academically and practically significant to reduce battery disposal-related negatives and drone operational costs through drone routing, recharging and fleet sharing decisions.

2.2. Blockchain-enabled sharing economy

The sharing economy was first introduced by Lessig (2008) and refers to a business model that consumes resources collaboratively through activities of sharing, exchanging and renting, while not owning the goods. Given the merits of a sharing economy in improving asset use rate, cost control ability and operational flexibilities, a growing number of studies from the transportation field have considered its application for operational improvements. For instance, Lu et al. (2017) considered a parking lot allocation problem with fleet sharing and repositioning. Kabra et al. (2020) proposed a structural choice model to analyse the effects of stations and availability of vehicles in a bike sharing system. He et al. (2020) analytically revealed insights for deploying a charging network and operating an electric vehicle sharing system. However, as Nesta (2015) stated, trust is the most critical currency to buy participants' shared resources on a sharing platform, yet, regardless of which best practices are adopted, it is never guaranteed that transactions between people will be trustworthy. To overcome this problem, blockchain is considered a trust-enhancement mechanism for a sharing economy.

The study by Greiner and Wang (2015) introduced the notion of ‘trust-free’ systems and called for the implementation of blockchain technology to automate an immutable, consensually agreed and publicly available record of past transactions that is governed by the whole system to mitigate trust issues in peer-to-peer systems. This can also help replace traditional contracts with a contractual agreement embedded in the process of the system itself. In this sense, costly mechanisms to build trust in intermediaries or interpersonal trust are rendered obsolete by design (Greiner and Wang, 2015). Later research by Fröwis and Böhme (2017) examined how such a smart contract structure contributes to a trust-free system through integrity over time. Pazaitis et al. (2017) used the illustrative case of Backfeed to reveal the potential of using blockchain to better support the dynamics of social sharing, and build a theoretical framework to assess and distribute social values. Previous research has also considered the use of blockchain to enhance platform connection ability for computational resource-sharing systems, such as grid computing and cloud computing (Hong et al., 2017; Stanciu, 2017; Teslya and Smirnov, 2018).

In all above studies, blockchain as a trust-enhancement solution for the sharing economy was mainly explored at a strategical or tactical level, while its implications at operational level were barely discussed. In particular, questions about how the performance of a specific application could be boosted by blockchain were not evaluated quantitatively, and how operational decisions should be adjusted accordingly were not answered. Hua et al. (2018) proposed a blockchain-based solution for battery refuelling. The lifetime information for all batteries was recorded in the blockchain network, so that owners of electric vehicles and station grids were guaranteed fair transactions for battery sharing. Although their research applied blockchain technology to construct a battery refuelling application, their study did not include how different operational decision making was affected accordingly.

Given the above research gaps, this paper aimed to address the challenges of drone routing problems through a blockchain-enabled fleet sharing solution. The key contributions are threefold. First, this is the first drone routing research considering fleet sharing and the effectiveness of using blockchain technology for sharing ability

enhancement. Second, this study considers the drone battery degradation effect together with routing and charging decisions. Third, this study is the first to consider the joint optimisation of operational costs and environmental effects.

3. Problem formulation

This section presents the problem in detail and proposes a mixed-integer programming formulation to achieve the research objectives. This approach will help operators explicitly minimise battery-related costs and other operational costs for sustainable supply chain development. The proposed model also considers battery-related features as constraints in formulating the problem.

3.1. Model description

This study considers a drone routing problem (DRP) with blockchain-enabled fleet sharing among multiple operators, which we define as a DRPBFS problem. In particular, operators in DRPBFS manage the same type of drones and serve their customers with the same type of products. Demands from customers are shared with each operator, and every single piece of demand is framed with a time window, which implies how early and how late each customer would like to accept delivery. To service customers within the given time window, every operator owns a depot in which commodities and drones are kept, and, from there, the operator must decide a cost-effective plan about which drone will be deployed for a specific demand with a certain route at a specific time point. Throughout all cost components, the composition of battery costs is complicated. Specifically, when a rechargeable battery runs from fully charged until the next recharging, this is labelled one charge cycle, and every lithium-ion battery has a cycle-to-failure (*CTF*) value. This value indicates the maximal number of cycles that a battery lasts and, once it is run out, operators must purchase a new battery for replacement, and the old one will be disposed. However, a drone's *CTF* value is not always fixed, but varies depending on the depth-of-discharge (*DOD*) level for one charge cycle. The *DOD* level indicates how much volume of electricity is used in a battery before it is charged. According to

Jeong et al. (2015), the *CTF* value of a battery should exponentially decrease alongside the increase of *DOD*—also known as a battery wear effect or battery degradation.

Therefore, although the cost of purchasing a new battery is fixed, the capital cost associated with each charge cycle could differ and is largely affected by the *DOD* level before the battery is charged again. In this regard, operators must balance the consumption of *CTF* and *DOD* value each time and make decisions considering the drone routing and battery wear. This is a complicated yet important issue, given that, even though draining a battery could allow more demands to be fulfilled in one flight, it accelerates the battery life-to-failure. Conversely, charging a battery with a high level of *DOD* reduces the use of one battery charge cycle, yet slows the capacity loss of the drone battery for a long run. This study aims to address these issues by introducing a blockchain-enabled fleet sharing platform. In this way, each drone does not need to return to the departure depot for reloading and battery charging. Instead, it can fly to other operators' depots to swap a fully-charged drone out to continue the work. This enables the shift from independent business operations to collaboration, and operators can integrate all information to optimise their decisions about which route will be used by the shared drones, when to recharge those drones and where to recharge them under the new context. Moreover, with respect to different sharing protocols, the trust level created by the corresponding sharing system is different. When operators have doubts about the shared resources (e.g., conditions of shared drones, if other operators have fully revealed their demand information), the level of their own resources to be shared with others will be discounted as well. When blockchain technology is incorporated as the sharing protocol, all information can be securely accessed and operators will have maximum willingness to share their resources. Hence, the maximum sharing ability will be provided by the underlying sharing platform.

Regarding above narrative, the following assumptions are made here to make the problem tractable:

1. Every customer is served only once.

2. The travel speed of each drone is assumed to be fixed.
3. The weight of each customer demand will be no greater than the maximal capacity of one drone.
4. Every depot owns the same type of drones and has the same commodities for service.
5. Each drone can serve multiple customers within one flight if its capacity allows.
6. Each drone can depart and land at any depot.
7. There is no need for a drone to have the same departure and arrival depot in one flight.
8. By the end of the overall planning horizon, the drone fleet size for each depot should remain the same as its initial amount.
9. Drones always depart from depots with a full battery.
10. The use of blockchain can avoid all trust issues and maximise the sharing ability of a sharing platform.

The assumptions are justified as follows. Assumptions 1 to 5 are commonly used in DRP-related literature (e.g., Chen et al., 2020). Although drone fleet sharing is barely discussed in the existing literature, this practice is not uncommon in collaborative VRP studies (Nataraj et al., 2019; Yao et al., 2019). Similar to research from Wang et al. (2020b), under a fleet sharing context, Assumptions 6 to 8 provide every operator equal access to the shared fleet and help form the foundation for collaboration. Assumption 9 reflects a common industrial practice. Assumption 10 is supported by different literature (e.g., Greiner and Wang, 2015; Pazaitis et al., 2017).

3.2. Notations and problem formulation

The notation below illustrates all the relevant sets, parameters and decision variables of the proposed model.

Notations

| Sets | |
|----------------|--|
| O | Set of depots (operators) |
| C | Set of customers |
| C^S | Subset of customers that can be shared with multi-depots |
| C^i | Subset of customers that can only be served by drones dispatched from the i -th depot node, $\forall i \in O$ |
| D | Set of drones |
| A | Set of arcs |
| N | Set of nodes |
| Indices | |
| i, j, l | Indices of node set N |
| (i, j) | Index of arc set A |
| d | Index of drone set D |
| Parameters | |
| G | Directed graph with node sets N and arc sets A , where $N = C \cup O$ and $A = (N, A) \{(i, j) \forall i, j \in N, i \neq j\}$ |
| m | Norm of the customer's set |
| n | Norm of the operator's set |
| \bar{T}_{ij} | Travel time for arc $(i, j), \forall (i, j) \in A$ |
| C_{ij} | Travel cost for arc $(i, j), \forall (i, j) \in A$ |
| Q_i | Demand of customer node $i, \forall i \in C$ |
| T_i^{LA} | Latest arrival time requested by customer node $i, \forall i \in C$ |
| T_i^{ED} | Earliest departure time allowed by customer node $i, \forall i \in C$ |
| K_i | Number of drones owned by depot node $i, \forall i \in O$ |
| B | Maximal load capacity of a drone |
| S | Average speed of a drone |
| S | Speed of a drone |
| E | Maximal flight duration of a drone |
| F | Purchase cost of a new battery |
| δ | Ratio of battery disposal cost to purchase cost |
| M | Sufficiently large number |
| State variable | |

| | |
|--------------------|---|
| y^d | Depth of discharge (<i>DOD</i>) of drone d before it is charged again, where $\forall d \in D$ |
| CTF | Drone battery cycle-to-failure value |
| e_i^d | Cumulative flight time when drone d arrives at customer node i , where $\forall i \in N, d \in D$ |
| t_i^d | Arrival time of drone d at customer node i , where $\forall i \in N, d \in D$ |
| v_i^d | Equals to 1 if the d -drone is dispatched from depot i ($\forall i \in O$), and 0 otherwise. |
| Decision variables | |
| x_{ij}^d | Drone d is deployed on arc (i, j) , where $\forall (i, j) \in A, d \in D$ |

The problem is first mapped out in a directed graph $G = (N, A)$, where $N = C \cup O$ and $A = \{(i, j) | i, j \in N, i \neq j\}$. In particular, node set N contains both customer and depot nodes and arc set A contains all node arcs. Further, the customers set is denoted as $C = \{1, 2, \dots, m\}$, and is further partitioned by $C = C^S \cup_{i \in O} C^i$. This indicates that subset C^S contains customers who can be served by shared resources, yet subset C^i contains customers who can only be served by the appointed depot i ($\forall i \in O$). The depot set is denoted $O = \{m + 1, m + 2, \dots, m + n\}$, where m is the size of customers (i.e., $m = |C|$) and n is the size of depots (i.e., $n = |O|$). With regard to each customer node i ($\forall i \in C$), Q_i represents its demand and is also associated with a hard time window $[T_i^{LA}, T_i^{ED}]$. For each depot node i ($\forall i \in O$), drones can be dispatched to service customers and returned for charging, and all available drones are denoted as set $D = \{1, 2, \dots, |D|\}$, where d is the index of this set (i.e., $\forall d \in D$). Benefitting from the blockchain-enabled fleet sharing, drones can be deployed and returned to any depot if that is more cost effective; however, for asset management consideration, every depot should have the same amount of drones as its initial stock by the end of the planning horizon. Further, the load capacity of each drone is B , and every single customer demand should be no more than a full drone load (i.e., $B \geq Q_i, \forall i \in C$). As stated by Assumption 2, drones will be travelling over each arc (i, j) ($\forall (i, j) \in A$) with speed S , and, given the triangle inequality rule, the shortest travel time between any two nodes, i and j , in a service area is always a straight Euclidean distance, $i \rightarrow j$. Therefore, the

corresponding travel time for arc (i, j) can be denoted by \bar{T}_{ij} (e.g., Chen et al., 2020) and the associated cost is C_{ij} . Without loss of generality, we also do not consider the service time of each customer, since it can be implicitly indicated by \bar{T}_{ij} as well. Accordingly, formulation of the original problem (OP) is given below:

$$\min \sum_{d \in D} \frac{(1 + \delta)F}{f_{CTF}(y^d)} \sum_{(i,j) \in A: i \in O} x_{ij}^d + \sum_{d \in D} \sum_{(i,j) \in A} C_{ij} x_{ij}^d \quad (1)$$

s.t.

$$f_{CTF}(y^d) = a_0 + a_1 y^{d-1} + a_2 y^{d-2} + a_3 y^{d-3}, \quad \forall d \in D \quad (2)$$

where $a_0 = -4,790$, $a_1 = 7,427$, $a_2 = -1,077$ and $a_3 = 55.4$, respectively:

$$\sum_{d \in D} \sum_{(i,j) \in A} x_{ij}^d = 1, \quad \forall i \in C \quad (3)$$

$$\sum_{d \in D} \sum_{(i,j) \in A} x_{ij}^d \leq K_i, \quad \forall i \in O \quad (4)$$

$$\sum_{(i,j) \in A: i \in O} x_{ij}^d \leq 1, \quad \forall d \in D \quad (5)$$

$$\sum_{(i,j) \in A} x_{ij}^d - \sum_{(j,i) \in A} x_{ji}^d = 0, \quad \forall i \in C, d \in D \quad (6)$$

$$\sum_{d \in D} \sum_{(i,j) \in A} x_{ij}^d - \sum_{d \in D} \sum_{(j,i) \in A} x_{ji}^d = 0, \quad \forall i \in O \quad (7)$$

$$v_i^d = x_{ij}^d, \quad \forall i \in O, j \in C, d \in D \quad (8)$$

$$x_{ji}^d \leq M(1 - v_i^d), \quad \forall i \in O, j, l \in C \setminus (C^S \cup C^i) \quad (9)$$

$$M(x_{ji}^d - 1) + \bar{T}_{ji} \leq t_i^d, \quad \forall j \in O, i \in C, d \in D \quad (10)$$

$$M(x_{ij}^d - 1) + t_i^d + \bar{T}_{ij} \leq t_j^d, \quad \forall i \in C, j \in N, d \in D \quad (11)$$

$$T_i^{LA} \leq t_i^d \leq T_i^{ED}, \quad \forall i \in C, d \in D \quad (12)$$

$$M(x_{ji}^d - 1) + \bar{T}_{ji} \leq e_i^d, \quad \forall j \in O, i \in C, d \in D \quad (13)$$

$$M(x_{ij}^d - 1) + e_i^d + \bar{T}_{ij} \leq e_j^d, \quad \forall i \in C, j \in N, d \in D \quad (14)$$

$$\frac{e_i^d}{E} \leq y^d \leq 1, \quad \forall i \in O, d \in D \quad (15)$$

$$\sum_{(i,j) \in A} Q_i x_{ij}^d \leq B, \quad \forall d \in D \quad (16)$$

$$0 \leq e_i^d \leq E, \quad \forall i \in O, d \in D \quad (17)$$

$$x_{ij}^d \in \{0,1\}, \quad \forall d \in D, (i,j) \in A \quad (18)$$

Eq. (1) presents the objective function of the OP model. It minimises the sum of two cost components: the first term is the sum of battery-related costs (i.e., new replacement costs and sustainable costs) for all drones over all arcs, and the second term is the sum of travelling costs for all drones over all arcs. To investigate the effects of battery charging decision on battery lifespan, Eq. (2) is a logarithmic polynomial function that describes the relationship (Fig. 1 further depicts the graph) between battery depth of discharge (*DOD*) and battery cycle-to-failure (*CTF*) (Jeong et al., 2015). Eq. (3) guarantees that every customer can only be served once. Eq. (4) enforces that the number of drones that can be dispatched from each depot does not exceed the total available amount. Eq. (5) indicates that not all drones should be used. Eq. (6) and (7) balance the flow at customer and depot nodes. Eq. (8) identifies the departure depot for a specific drone. Eq. (9) forbids the drone from visiting the unshared customer nodes belonging to other depots. By introducing a sufficiently large constant M , we can linearise the ‘if-then’ constraint—that is, if the d -drone is dispatched from depot i , then the d -drone does not visit customer l . In practice, the domains of the associated variables can be used for calibrating the value of the big M . For constraints (9), the domains of variables x_{jl}^d and v_i^d belong to $\{0,1\}$, so the upper bounds of the variables are both 1. We can set the value of the M to any number that is greater than or equal to 1, such as 1. This ensures that the ‘if-then’ constraints will not be violated. For constraints (10), they can be transformed as $\bar{T}_{ji} - t_i^d \leq M(1 - x_{ji}^d)$, $\forall j \in O, i \in C, d \in D$. Since $t_i^d \geq T_i^{LA}$ and $x_{ji}^d \in \{0,1\}$, so if we set M satisfying $\max_{\forall j \in O, i \in C} \{\bar{T}_{ji} - T_i^{LA}\} \leq M$, the ‘if-then’ constraints will not be violated. For

constraints (11), they can be transformed as $t_i^d + \bar{T}_{ij} - t_j^d \leq M(1 - x_{ij}^d)$, $\forall i \in C, j \in N, d \in D$. Since $t_i^d \geq T_i^{LA}$, $t_j^d \leq T_j^{ED}$, and $x_{ij}^d \in \{0,1\}$, so if we set M satisfying $\max_{\forall i \in C, j \in N} \{\bar{T}_{ij} - T_i^{LA} + T_j^{ED}\} \leq M$, the ‘if-then’ constraints will not be violated. Eq. (10) to (12) ensure that the time window of each customer will not be violated. Eq. (13) and (14) illustrate the flow conservation in terms of the drone’s energy. The values of the M here is similar to that from constraints (10)-(11). For constraints (13), M needs to satisfy $\max_{\forall j \in O, i \in C} \{\bar{T}_{ji}\} \leq M$. For constraints (14), M needs to satisfy $\max_{\forall i \in C, j \in N} \{\bar{T}_{ij} + E\} \leq M$. Eq. (15) specifies the value range of DOD for a drone battery. Eq. (16) ensures that each load of a drone is within its maximal capacity. Eq. (17) and (18) define the domains of the decision variables.

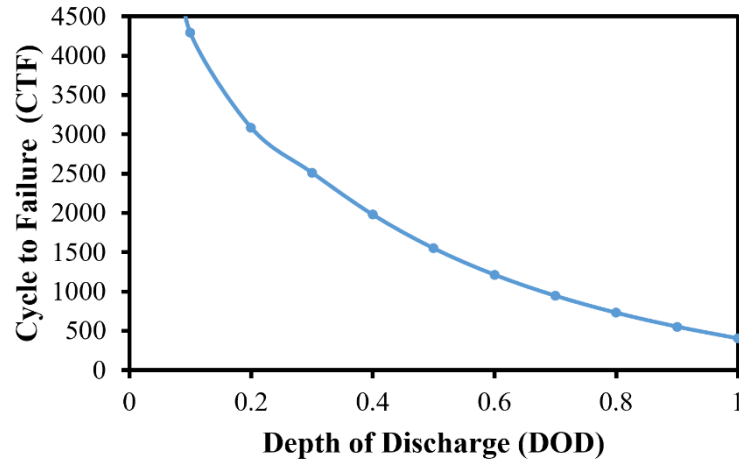


Figure 1. Relationship between CTF and DOD

Proposition 1. The DRPBFS is an NP-complete problem.

Proof. The current problem can be reduced to a vehicle routing problem with time windows (VRPTW) if the network only contains one depot and the deployed drones have no flight duration constraints. This completes the proof.

4. Solution strategy

This section introduces the methodology used to solve the problem. As an NP-complete problem (Proposition 1), DRPBFS is difficult to solve directly by commercial optimisation software, such as IBM CPLEX or Gurobi, especially when the scale of the problem is relatively large. To enhance the practicability of this study, a new method has been proposed that can perform more efficiently in solving the problem.

4.1. OP reformulation

Given the battery wear effect, the corresponding cost function brings a non-linear term into the objective function of the OP formulation. To simplify the solution process, we first reformulate OP to remove the non-linear component. In particular, we introduce a new set of auxiliary variables, z^d ($d \in D$), to replace the non-linear part, $\sum_{d \in D} \frac{(1+\delta)F}{f_{CTF}(y^d)} \sum_{(i,j) \in A: i \in O} x_{ij}^d$, in the OP formulation. To ensure the new formulation is equivalent to OP, a series of new constraints are added based on the McCormick envelope (Lundell et al., 2009). The details of this new formulation are presented below.

P2:

$$\min \sum_{d \in D} z^d + \sum_{d \in D} \sum_{(i,j) \in A} c_{ij} x_{ij}^d \quad (19)$$

$$\text{Eq. (3) - (17)} \quad (20)$$

$$z^d \leq M \sum_{(i,j) \in A: i \in O} x_{ij}^d, \forall d \in D \quad (21)$$

$$\frac{(1+\delta)F}{f_{CTF}(y^d)} - M \left(1 - \sum_{(i,j) \in A: i \in O} x_{ij}^d \right) \leq z^d \leq \frac{(1+\delta)F}{f_{CTF}(y^d)}, \quad \forall d \in D \quad (22)$$

$$z^d \geq 0, \quad \forall d \in D \quad (23)$$

Eq. (20) is a copy of the constraints from OP. With the newly added Eq. (21) and (22), the new formulation P2 no longer contains any non-linear term. For constraints (21),

as $z^d \leq \frac{(1+\delta)F}{f_{CTF}(y^d)}$ and $x_{ij}^d \in \{0,1\}$, if we set M satisfying $\max_{0 \leq y^d \leq 1} \left\{ \frac{(1+\delta)F}{f_{CTF}(y^d)} \right\} \leq M$, the

‘if-then’ constraints will not be violated. For constraints (22), as $z^d \geq 0$ and $x_{ij}^d \in$

$\{0,1\}$, if we set M satisfying $\max_{0 \leq y^d \leq 1} \left\{ \frac{(1+\delta)F}{f_{CTF}(y^d)} \right\} \leq M$, the ‘if-then’ constraints will not be violated. Eq. (23) defines the domains of z^d .

4.2. Branch-and-price algorithm

Based on P2, we further reformulate it to a path-based model that includes both the master problem (MP) and sub-problem (SP), so a branch-and-price-based algorithm can be developed.

4.2.1. Dantzig-Wolfe (DW) decomposition

In P2, only Eq. (3), (4) and (7) (from OP originally) constrain to all drones. Based on the principle of DW decomposition, they are the first selection to be decomposed for a path-based reformulation (P3). In DRPBFS, the drone path is defined as a path with start and end at the depot (not necessarily the same one), and a series of customers in between. In this sense, the MP of P3 can be first obtained with the support of some new parameters and decision variables:

| Sets | |
|--------------------|---|
| Ω | Set of feasible drone paths, indexed by k |
| Ω_1 | Subset of Ω |
| Parameters | |
| c^k | Travel cost of drone path k , $\forall k \in \Omega$ |
| a_{ij}^k | Equals 1 if path k goes through the arc $(i,j) \in A$, and 0 otherwise |
| z^k | Battery-related cost for path k , $\forall k \in \Omega$ |
| Decision variables | |
| θ_k | Equals 1 if path k is selected, and 0 otherwise |

P3-MP:

$$\min \sum_{k \in \Omega} (z^k + c^k) \cdot \theta_k \quad (24)$$

s. t.

$$\sum_{k \in \Omega} \sum_{(i,j) \in A} a_{ij}^k \theta_k \geq 1, \quad \forall i \in C \quad (25)$$

$$\sum_{k \in \Omega} \sum_{(i,j) \in A} a_{ij}^k \theta_k \leq K_i, \quad \forall i \in O \quad (26)$$

$$\sum_{k \in \Omega} \sum_{(i,j) \in A} a_{ij}^k \theta_k - \sum_{k \in \Omega} \sum_{(j,i) \in A} a_{ji}^k \theta_k = 0, \quad \forall i \in O \quad (27)$$

$$\theta_k \in \{0,1\}, \quad \forall k \in \Omega \quad (28)$$

Eq. (24) is the objective function that minimises both travelling- and battery-related costs for all selected paths. Eq. (25) indicates that every demand should be served. To accelerate the convergence, it is defined as an inequality to loosen constraint. Yet each customer will not be visited more than once, as it is constrained by the objective function and triangle rule. Eq. (26) constrains the maximum number of drones that can be used in each depot. Eq. (27) represents the flow conservation at each depot. Eq. (28) defines the domain of the decision variable θ_k . By relaxing θ_k as a continuous variable and allowing set Ω_1 to be the subset of Ω , we can further obtain a linearly relaxed restricted master problem (RMP). In addition, we choose the nearest neighbour (NN) heuristic to generate the initial solution for the RMP.

| Parameters | |
|--------------------|---|
| \bar{c}_{ij} | Reduced cost of arc $(i,j) \in A$ |
| Decision variables | |
| z | Battery-related cost for drone path |
| t_i | Time when drone arrives at customer i |
| e_i | Cumulative flight time when drone arrives at customer i |
| y | Depth of discharge (<i>DOD</i>) of drone |
| s_{ij} | Equals 1 if the arc $(i,j) \in A$ is selected, and 0 otherwise |
| State variable | |
| v_i | Equals 1 if depot is the departure for a route, and 0 otherwise |

Next, we present the pricing strategy to formulate the sub-problem (P3-SP) with the support of the above parameters and decision variables. Let λ , μ and ν be the dual vectors that are related to Eq. (25) to (27) (the corresponding dual model is in Appendix B). Then the reduced cost of a drone path is:

$$\bar{c} = z + \sum_{(i,j) \in A: i \in O} (C_{ij} - \mu_i - \nu_i) a_{ij}^k + \sum_{(i,j) \in A: i, j \in C} (C_{ij} - \lambda_i) a_{ij}^k + \sum_{(i,j) \in A: j \in O} (C_{ij} + \nu_j - \lambda_i) a_{ij}^k \quad (29)$$

If we exclude the battery cost z from the above equation, the reduced cost of a drone path can be further decomposed based on arcs (see below Eq. [30]):

$$\bar{c}_{ij} = \begin{cases} C_{ij} - \mu_i - \nu_i, & i \in O \\ C_{ij} - \lambda_i, & i, j \in C \\ C_{ij} + \nu_j - \lambda_i, & j \in O \end{cases} \quad (30)$$

For a given arc (i, j) from a drone path, \bar{c}_{ij} represents an integrated form of the reduced costs (with z excluded) based on the types of its corresponding arc nodes. If this is used to replace the second, third and fourth term in Eq. (29), the sub-problem of P3 can be obtained as below:

P3-SP:

$$\min z + \sum_{(i,j) \in A} \bar{c}_{ij} s_{ij} \quad (31)$$

s.t.

$$\sum_{(i,j) \in A: i \in O} s_{ij} = \sum_{(j,i) \in A: i \in O} s_{ji} = 1 \quad (32)$$

$$\sum_{(i,j) \in A} s_{ij} - \sum_{(j,i) \in A} s_{ji} = 0, \quad \forall i \in C \quad (33)$$

$$v_i = s_{ij}, \forall i \in O, j \in C \quad (34)$$

$$s_{jl} \leq M(1 - v_i), \forall i \in O, j, l \in C \setminus (C^S \cup C^I) \quad (35)$$

$$M(1 - s_{ji}) + \bar{T}_{ji} \leq t_i, \quad \forall j \in O, i \in C \quad (36)$$

$$M(1 - a_{ij}^k) + t_i^d + \bar{T}_{ij} \leq t_j, \quad \forall i \in C, j \in N \quad (37)$$

$$T_i^{LA} \leq t_i \leq T_i^{ED}, \quad \forall i \in C \quad (38)$$

$$M(s_{ij} - 1) + \bar{T}_{ji} \leq e_i, \quad \forall j \in O, i \in C \quad (39)$$

$$M(s_{ij} - 1) + e_i + \bar{T}_{ij} \leq e_j, \quad \forall i \in C, j \in N \quad (40)$$

$$0 \leq e_i \leq E, \quad \forall i \in O \quad (41)$$

$$\frac{e_i}{E} \leq y \leq 1, \quad \forall i \in O \quad (42)$$

$$\sum_{(i,j) \in A} Q_i s_{ij} \leq B \quad (43)$$

$$z \leq M \sum_{(i,j) \in A: i \in O} s_{ij} \quad (44)$$

$$\frac{(1+\delta)F}{f_{CTF}(y)} - M \left(1 - \sum_{(i,j) \in A: i \in O} s_{ij} \right) \leq z \leq \frac{(1+\delta)F}{f_{CTF}(y)} \quad (45)$$

$$z \geq 0 \quad (46)$$

$$s_{ij} \in \{0,1\}, \forall (i,j) \in A \quad (47)$$

Lemma 1. The battery cost z for a drone path is $\frac{(1+\delta)F}{f_{CTF}(y)}$.

Proof. With the help of Eq. (32), the component $\sum_{(i,j) \in A: i \in O} s_{ij}$ in both Eq. (44) and (45) can be dropped and replaced by 1. Therefore, Eq. (44) and (45) will be equivalent to the following inequality:

$$\frac{(1+\delta)F}{f_{CTF}(y)} \leq z \leq \frac{(1+\delta)F}{f_{CTF}(y)} \quad (48)$$

Eq. (48) is further equivalent to $z = \frac{(1+\delta)F}{f_{CTF}(y)}$. This completes the proof. Hence, P3-SP

can be rewritten as:

P4-SP:

$$\min \frac{(1+\delta)F}{f_{CTF}(y)} + \sum_{(i,j) \in A} \bar{c}_{ij} s_{ij} \quad (49)$$

s.t.

$$\text{Eq. (32) - (43), (46) - (47)} \quad (50)$$

Eq. (49) is the objective function that minimises the reduced cost of a column for a drone path. Eq. (50) copies the constraints for the initial formulation of P3-SP. In particular, Eq. (32) indicates that a drone can be dispatched from and returned to any depot. Eq. (33) ensures the flow conservation at each customer node. Eq. (34) and (35) forbid the route from containing a departure depot with unshared customer nodes belonging to depots. Eq. (36) to (38) enforce the satisfaction of the time window

requirement at each customer node. For constraints (35)-(37), the setting of the values of M is the same as that of constraints (9)-(11), respectively. They need to satisfy $1 \leq M$, $\max_{\forall j \in O, i \in C} \{\bar{T}_{ji} - T_i^{LA}\} \leq M$, and $\max_{\forall i \in C, j \in N} \{\bar{T}_{ij} - T_i^{LA} + T_j^{ED}\} \leq M$, respectively. Eq. (39) to (43) ensure the feasibility of flight duration, load capacity and DOD . For constraints (39)-(40), the setting of the values of M is the same as that of constraints (13)-(14) respectively. They need to satisfy $\max_{\forall j \in O, i \in C} \{\bar{T}_{ji}\} \leq M$ and $\max_{\forall i \in C, j \in N} \{\bar{T}_{ij} + E\} \leq M$ respectively. For constraints (44)-(45), the setting of the values of M is the same as that of constraints (21)-(22), i.e., $\max_{0 \leq y^d \leq 1} \left\{ \frac{(1+\delta)F}{f_{CTF}(y^d)} \right\} \leq M$. Eq. (46) and (47) define the domains of the decision variables.

4.2.2. Branch-and-price solution for P3-MP

With the above re-reformulation, we further introduce a branch-and-price algorithm to solve P3-MP. The structure of this algorithm is given as follows:

Step 1: Solve the RMP with the initial columns obtained by the NN heuristic.

Step 2: Solve P4-SP (see Section 4.2.3) to generate columns with negative reduced cost. Notably, P4-SP is not required to be solved to optimality in every iteration, since any column with negative reduced cost can improve the objective function of RMP. Therefore, whenever a negative column is found, it must be added to RMP for iteration. Otherwise, go to Step 3.

Step 3: If an integer solution is obtained, the algorithm is terminated and the integer solution is returned as the optimal solution. Otherwise, a branch-and-bound approach (see Section 4.2.4) will be further used to search for an integer solution by iterating the above introduced column generation process.

4.2.3. Labelling algorithm for P4-SP

To solve the sub-problem (i.e., P4-SP), P4-SP can be simplified to a classical elementary shortest path problem with resource constraints (ESPPRC), which is NP-hard (Garey and Johnson, 1979). Then, a dynamic programming algorithm named

the ‘labelling algorithm’ can be applied (e.g., Feillet et al., 2004; Righini and Salani, 2006). Unlike a general form of an ESPPRC, the resources cumulative flight time, e_i , affects the objective function of P4-SP and gives an additional reduced cost of a column for the drone path, which requires different handling techniques. Based on the labelling algorithm implemented by Desrochers (1988), we develop the following process to fit our sub-problem settings.

(1) Label structure

This is used to record the up-to-node i partial path and label it with a series of information (e.g., resource consumption information). In our problem, the label structure is expressed as Eq. (51):

$$L_i = \{i, predecessor(L_i), r(L_i), v(L_i), t(L_i), e(L_i), q(L_i)\} \quad (51)$$

In the above equation, i is the last node in the partial path; $predecessor(L_i)$ is the parental label, which extends to the current L_i ; and $r(L_i)$ is the accumulated reduced cost of L_i , which includes two cost components from Eq. (31). In particular, we define the first reduced cost component as $g(e(L_i))$ and the second as $c(L_i)$. $v(L_i)$ is a vector that performs as a tabu list that records visited nodes and un-accessible nodes (Feillet et al., 2004), $t(L_i)$ is the arrival time of the drone at node i , $e(L_i)$ is the accumulated flight time of the drone at node i , and $q(L_i)$ is the accumulated weight of customer parcels for the partial path.

(2) Label extension process

To begin this process, we start with a label L_i with the minimum reduced cost. A new label L_j is obtained by extending L_i through arc (i, j) if the time window and resources-based constraints are satisfied. The relationship between L_i and L_j is shown as follows:

$$t(L_j) = \max[t(L_i) + \bar{T}_{ij}, T_j^{ED}] \quad (52)$$

$$e(L_j) = e(L_i) + \bar{T}_{ij} \quad (53)$$

$$q(L_j) = q(L_i) + Q_j \quad (54)$$

$$c(L_j) = \begin{cases} c(L_i) + c_{ij} - \mu_i - \nu_i, & i \in O \\ c_{ij} - \lambda_i & i, j \in C \\ c_{ij} - \lambda_i + \nu_j & j \in O \end{cases} \quad (55)$$

$$\text{predecessor}(L_j) = L_i \quad (56)$$

$$r(L_j) = g(e(L_j)) + c(L_j) \quad (57)$$

Note that the update rules for $v(L_j)$ also consider the time window and demands of the next node, h , where node j will extend to. If node h is not accessible, it then should be updated to an un-accessible node set H , and this set is further included into $v(L_j)$:

$$H = \{h \mid t(L_j) + \bar{T}_{jh} > T_h^{LA}, e(L_j) + \bar{T}_{jh} > E, q(L_j) + Q_h > L\} \quad (58)$$

$$v(L_j) = v(L_i) \cup H \cup i \quad (59)$$

(3) Dominance rule

To avoid some unprofitable labels, the following dominance rule is elaborated. The two labels, $L_i^1 = \{i, \text{predecessor}(L_i^1), v(L_i^1), t(L_i^1), e(L_i^1), q(L_i^1)\}$ and $L_i^2 = \{i, \text{predecessor}(L_i^2), r(L_i^2), v(L_i^2), t(L_i^2), e(L_i^2), q(L_i^2)\}$, represent two different partial paths with the same last node i . If the following conditions (Desrochers, 1988) are satisfied, L_i^1 can dominate L_i^2 :

- lower reduced cost: $r(L_i^1) \leq r(L_i^2)$
- less visited nodes: $v(L_i^1) \subseteq v(L_i^2)$
- earlier arriving time: $t(L_i^1) \leq t(L_i^2)$
- lower cumulative flight time: $e(L_i^1) \leq e(L_i^2)$
- lower cumulative parcel weight: $q(L_i^1) \leq q(L_i^2)$.

If a label is dominated by other labels, the extension process for this label should be terminated and this label should be deleted.

4.2.4. Branching strategy

If a fractional solution is obtained from the above column generation, a branch-and-bound (BB) process must be performed. Given that the path is difficult to branch, we use the method proposed by Danna and Le Pape (2005) to select arcs for BB instead. Specifically, suppose x^* is the optimal solution of the RLMP at the current searching node in the BB tree. If x^* is a integer, then branching is not necessary. Otherwise, we iterate all paths with the fractional values and select the arc (i,j) with the largest value of $\sum_{k \in \Omega_1} x_{ij}$ to be branched on. In addition, given that adding additional constraints to MP will result in increased dual prices, we branch the fractional arcs through modifying their associated reduced costs indirectly (Degraeve and Jans, 2007). To search the BB tree, we incorporate the depth-first method to effectively obtain feasible solutions at leaf nodes for pruning. In particular, the search starts from the root node in one direction. If the target solution is found, then the process in this direction can be terminated. Otherwise, the previous node will be used to start the same process again in a different direction. The BB procedure is shown below:

- Step 1: If the solution does not yield an integer solution at the searching node, select one of the arcs (i,j) to branch on, and go to Step 2.
- Step 2: After selecting the arc (i,j) , make two branches according to the following rules:
 - i. The first child node: For the arc (i,j) that must be served (defined as $(i,j) = 1$) by drones, if both i and j are customer nodes, arcs (i,l) ($\forall l \in C, l \neq j$) and (l,j) ($\forall l \in C, l \neq i$) should be deleted so that all remaining routes that visit node i or j must pass through arc (i,j) . This will differ slightly when either i or j is the depot. We cannot delete the arcs visiting and leaving the node because the depot can be visited more than once.

- ii. The second child node: For the arc (i, j) that cannot be served (defined as $(i, j) = 0$) by drones, we only need to delete those arcs that should be imposed with $(i, j) = 0$ arcs.

After deleting the arcs according to the rules above, the corresponding routes must also be deleted from the column sets of RLMP.

5. Numerical experiment

In this section, a series of numerical experiments are designed to demonstrate both the methodological and managerial significance of our research. Based on use of a piecewise linear approximation with Gurobi (PLA-Gurobi), we conduct a preliminary test to examine the quality of its solution against an enumeration with very small size instances. We then benchmark the computation performance of our BP-based solution against the piecewise linear approximation with Gurobi for problems with various scales. Further, to explore the effects of battery degradation on drone operations, we conduct our experiments with different objective functions (i.e., with or without battery-related cost optimisation). Finally, to uncover more significant insights, different sensitivity analyses are undertaken, varying both economic and operational parameters. All experiments are performed on a Windows 10 OS with four 64-bit 3.4 GHz Intel Core processors and 64 GB RAM. The model is programmed by Java and the solver is Gurobi 9.0.0.

5.1. Computational performance benchmarking

5.1.1. Dataset and parameter settings

Instances used for all our experiments are generated with reference to Hexa-B Hexacopter fleet (Dorling et al., 2017). We set a new battery purchase cost (F) as \$500 and the battery disposal-related costs as twice that of a new purchase (i.e., $\delta = 2$). The load capacity of a drone (L) is 3 kg, and the maximal flight duration (E) is 50 minutes at a speed of $v = 6$ m/s. At the beginning of our experiments, unshared customer nodes are not considered (i.e., $C^i = \emptyset, \forall i \in O$).

To mark the position of each node, we introduce a coordinate system with its origin at (0,0), and the base unit for distance is kilometres. In the preliminary test, the operation is only performed between two depots, with coordinates of (-3,0) and (3,0), respectively. The service coverage of each depot is a circle with a radius of 5 km. We only consider scenarios with two and three customers within each service coverage for the preliminary test to simplify the situation. The relative position for one customer of a depot is defined by $[l*\sin\alpha, l*\cos\alpha]$, where l represents a direct distance between this customer to its depot and α is the angle of this customer to the x-axis. Also, l is uniformly distributed between $[0, 5]$, and α is uniformly distributed between $[0, 2\pi]$. Fig. 2 below illustrates the above configuration for one scenario. The initial drone fleet size is the same for all depots. For different instances in the preliminary test, we increment the drone fleet size for every depot from two to five drones for the preliminary test. The overall planning horizon is two hours and each customer associated time window is generated following these rules: the latest arrival time requested for customers is generated uniformly from $[0,120]$, then uniformly sampled from $[20,50]$ and added to the latest arrival time to attain the earliest departure time. Customer demands are uniformly generated from $\{0.5 \text{ kg}, 1 \text{ kg}, 1.5 \text{ kg}, 2 \text{ kg}\}$. To easily refer to a specific experiment, we use a code format A##-##-# for referencing an instance. In particular, A## represents the number of depots considered for the underlying instance. The following ## represents the number of customers serviced by each depot. The last # is a sample ID for instances sharing the same scale.

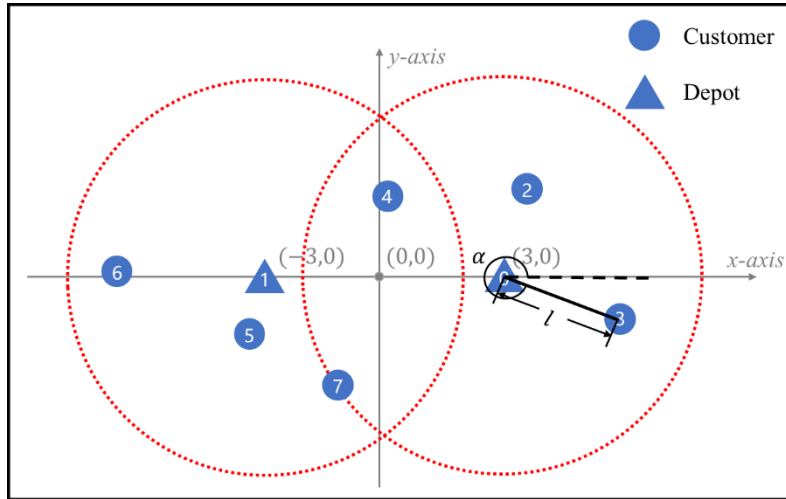


Figure 2. Example of customer and depot distribution for preliminary test

For experiments with larger instances, we consider scenarios with two, three, four and five depots, respectively. These depots are randomly located on a circle with a radius of 5 km and its centre at the origin of the coordinate system. Each depot's service coverage is the same as above, yet covers five to 20 customers, depending on the instance. An example of such a setting is illustrated in Fig. 3 below. In addition, for different instances, the number of drones in each depot can range from four to 10. All other settings remain the same as in the preliminary test.

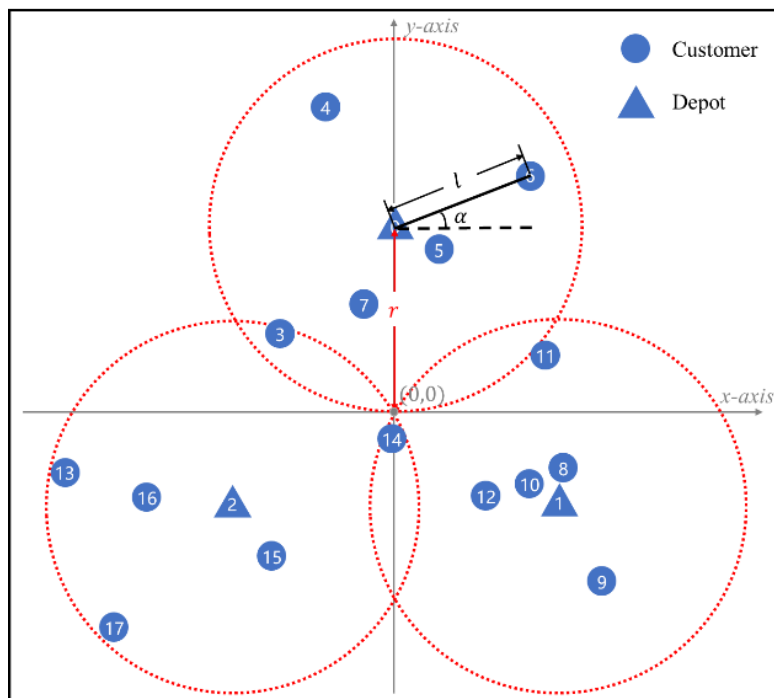


Figure 3. Example of customer and depot distribution for large instance

5.1.2. Preliminary test

Given the non-linear component—that is, the battery-combined objective function of the OP model—the OP cannot be solved directly via the off-the-shelf solver. To evaluate the solution quality and computational performance of the proposed BP algorithm, we derive an enumeration-based method (see details in Appendix C) and the PLA-Gurobi (see details in Appendix D), respectively, for benchmarking. To explore the solution quality of PLA-Gurobi, the enumeration method is used as a baseline for benchmarking. To make the enumeration method tractable, we begin the experiment on a very small scale. Table 1 below presents the experimental results.

Table 1. Results on small instances (K = number of drones; Opt. = optimality; CPU = computation time)

| Ins. | Enumeration | | | PLA-Gurobi | | | | Branch-and-price | | | |
|--------|-------------|--------------|------------|------------|--------------|--------|--------------|------------------|--------------|-------|-----|
| | K | Obj. | CPU | K | Obj. | CPU | Gap | K | Obj. | CPU | Gap |
| A2-2-1 | 2 | 1.726 | 2.020 | 2 | 1.726 | 0.396 | 0 | 2 | 1.726 | 0.085 | 0 |
| A2-2-2 | 3 | 2.257 | 1.369 | 3 | 2.257 | 0.334 | 0 | 3 | 2.257 | 0.069 | 0 |
| A2-2-3 | 3 | 2.190 | 1.631 | 3 | 2.190 | 0.270 | 0 | 3 | 2.190 | 0.076 | 0 |
| A2-2-4 | 2 | 1.419 | 1.989 | 2 | 1.419 | 0.288 | 0 | 2 | 1.419 | 0.083 | 0 |
| A2-2-5 | 2 | 1.881 | 1.924 | 3 | 2.067 | 0.374 | 9.89% | 2 | 1.881 | 0.083 | 0 |
| A2-3-1 | 5 | 4.876 | 1,660.471 | 5 | 4.876 | 0.599 | 0 | 5 | 4.876 | 0.081 | 0 |
| A2-3-2 | 4 | 4.381 | 4,750.790 | 4 | 4.381 | 57.414 | 0 | 4 | 4.381 | 0.083 | 0 |
| A2-3-3 | 3 | 2.845 | 13,220.682 | 3 | 2.845 | 8.286 | 0 | 3 | 2.845 | 0.088 | 0 |
| A2-3-4 | 5 | 4.803 | 3,699.803 | 5 | 4.862 | 1.070 | 1.23% | 5 | 4.803 | 0.092 | 0 |
| A2-3-5 | 4 | 4.048 | 4,191.452 | 3 | 4.132 | 1.609 | 2.08% | 4 | 4.048 | 0.202 | 0 |

In Table 1, the results from enumeration provide the baseline for benchmarking. As a solution to an NP-complete problem, the CPU time of the enumeration approach increases exponentially alongside the increase of problem scale. Comparatively, the use of piecewise approximation and BP is much more efficient. For the comparison of the objective values: (1) the BP algorithm, as an exact solution, produces exactly the same results as the enumeration and (2) a high level of accuracy for the results is obtained from PLA-Gurobi, given that seven results produced have zero gap to enumeration and the other three have relatively small gaps. Taking PLA-Gurobi's solution quality and computational efficiency into account, PLA-Gurobi is used as the benchmark in the next experiments with larger instances. In particular, we use different problem scales to further evaluate our solution's computational performance in efficiency.

5.1.3. Experiments for medium-scale instances

This section examines scenarios including two and three depots. Accordingly, each depot’s number of customers ranges from five to 10 in each experiment, and, in total, this yields 15 instances explicitly. The PLA-Gurobi and BP algorithm are both implemented to solve these instances. We also set 7,200 seconds as the time tolerance for each instance. The details of the results are presented in Table 2.

Table 2. Computational results for 15 instances (LB = lower bound at root node; Gap = % diff. of opt. to Gurobi)

| Ins. | Gurobi | | | | Branch-and-price | | | | Gap |
|---------|--------|---------|---------------|--------------|------------------|--------|---------------|-------|-------|
| | K | LB | Opt. | CPU | K | LB | Opt. | CPU | |
| A2-5-1 | 6 | 2.071 | 6.167 | 21.895 | 5 | 6.023 | 6.082 | 0.125 | 1.40% |
| A2-5-2 | 4 | 1.654 | 5.909 | 381.671 | 4 | 5.843 | 5.909 | 0.14 | 0 |
| A2-5-3 | 5 | 2.121 | 6.103 | 54.447 | 6 | 6.068 | 6.068 | 0.094 | 0.58% |
| A2-5-4 | 5 | 1.346 | 4.208 | 9.337 | 5 | 4.062 | 4.071 | 0.112 | 3.37% |
| A2-5-5 | 5 | 1.58 | 6.135 | 92.155 | 5 | 5.804 | 6.135 | 0.251 | 0 |
| A2-10-1 | 10 | 3.815 | 12.477 | $\geq 7,200$ | 10 | 12.308 | 12.477 | 0.565 | 0 |
| A2-10-2 | 9 | 3.32497 | 11.527 | $\geq 7,200$ | 9 | 11.312 | 11.455 | 0.287 | 0.63% |
| A2-10-3 | 10 | 3.057 | 10.884 | $\geq 7,200$ | 10 | 10.802 | 10.802 | 0.234 | 0.76% |
| A2-10-4 | 9 | 3.477 | 11.025 | $\geq 7,200$ | 9 | 11.025 | 11.025 | 0.187 | 0 |
| A2-10-5 | 10 | 3.521 | 12.317 | $\geq 7,200$ | 10 | 11.759 | 11.758 | 0.147 | 4.75% |
| A3-5-1 | 9 | 2.411 | 8.099 | $\geq 7,200$ | 9 | 7.775 | 8.099 | 0.205 | 0 |
| A3-5-2 | 9 | 2.826 | 9.527 | $\geq 7,200$ | 9 | 9.406 | 9.420 | 0.171 | 1.14% |
| A3-5-3 | 8 | 2.809 | 8.917 | 1,606.245 | 9 | 8.766 | 8.816 | 0.156 | 1.15% |
| A3-5-4 | 8 | 2.827 | 9.408 | $\geq 7,200$ | 8 | 9.404 | 9.408 | 0.187 | 0 |
| A3-5-5 | 11 | 3.4 | 11.204 | $\geq 7,200$ | 11 | 11.005 | 11.204 | 0.205 | 0 |

Regarding the results, all 15 instances are solved to optimality by the BP algorithm in less than 0.3 seconds. Comparatively, PLA-Gurobi can only solve seven of 15 instances within our time tolerance (i.e., 7,200 seconds), and, even for the solved instances, some can consume over hundreds of seconds.

Compared with the initial LB, we find that the LB obtained by PLA-Gurobi is far worse than the BP algorithm, which spends a great deal of time for convergence. In addition, for instances such as A2-10-1 and A2-10-4, even though PLA-Gurobi finds the optimal solution, the gap between its UB and LB is more than 30% when its computation time reaches tolerance. This can be explained as the LB obtained from the relaxation of the OP model is worse than that obtained from RMP (Song et al., 2017). To further evaluate the BP algorithm’s computational capability, our next section further enlarges the experimental instance scales.

5.1.4. Experiments for large-scale instances

Given the computational limit of PLA-Gurobi (reflected in Section 5.1.3), it is excluded from the following large-scale experiments. Instead, only the BP algorithm is implemented for the experiments below (in Table 3). In particular, we examine scenarios with four and five depots. Each depot services 15 or 20 customers, and, in total, 20 different scenarios are explored.

Table 3. Computational results of BP algorithm on large-scale instances

| Ins. | K | LB | Opt. | CPU | Ins. | K | LB | Opt. | CPU |
|---------|----|--------|--------|---------|---------|----|--------|--------|------------|
| A4-15-1 | 28 | 26.761 | 26.960 | 4.402 | A5-15-1 | 36 | 33.136 | 33.535 | 55.859 |
| A4-15-2 | 28 | 25.019 | 25.019 | 1.917 | A5-15-2 | 33 | 34.121 | 34.473 | 1,536.446 |
| A4-15-3 | 23 | 22.908 | 23.140 | 15.722 | A5-15-3 | 34 | 30.635 | 31.069 | 594.552 |
| A4-15-4 | 27 | 27.226 | 27.355 | 1.841 | A5-15-4 | 34 | 34.320 | 34.684 | 62.398 |
| A4-15-5 | 30 | 28.674 | 29.057 | 54.131 | A5-15-5 | 34 | 35.958 | 36.367 | 308.033 |
| A4-20-1 | 35 | 33.954 | 34.562 | 256.668 | A5-20-1 | 48 | 40.250 | 40.753 | 469.188 |
| A4-20-2 | 34 | 35.688 | 36.098 | 280.542 | A5-20-2 | 45 | 40.364 | 40.829 | 711.219 |
| A4-20-3 | 36 | 35.361 | 35.990 | 493.538 | A5-20-3 | 43 | 43.716 | 44.013 | 5,018.505 |
| A4-20-4 | 36 | 39.078 | 39.448 | 7.807 | A5-20-4 | 43 | 41.639 | 42.131 | 2,972.804 |
| A4-20-5 | 35 | 36.957 | 37.235 | 922.841 | A5-20-5 | 43 | 42.648 | 43.184 | 11,726.634 |

The results on large-scale instances obtained by the BP algorithm are shown in Table 3. For all instances, the BP algorithm can obtain optimality for a gap (i.e., $\frac{(Opt.-LB)}{Opt.} \times 100\%$) of zero. Alongside the increase in instance size, the computational time of the BP algorithm increases greatly. However, its practicability is greatly improved, as it can obtain an optimal solution for a problem with 100 nodes (i.e., from A5-20-1 to A5-20-5) within an average of 5,000 seconds.

5.2. Comparison of DRPBFS and DRP

To investigate the operational effectiveness of the blockchain-enabled fleet sharing, we compare the DRPBFS with DRP (i.e., drone routing without fleet sharing) under different instances. Specifically, the two cost components (travel costs and battery-related costs) in our objective function are discussed explicitly. Table 4 below details the experimental results comparing travel costs, battery-related costs and total costs for the two models under different instances. In addition, when a flight has a different departure depot from the arrival one, it indicates this flight is completed by a shared drone. In this sense, we introduce an indicator named ‘ST rate’ to record the percentage of such flights out of the total flights for each instance. This can be further used to explore how the level of fleet sharing frequency would be affected by different operational factors.

As demonstrated by the below results, DRPBFS has a cost-effective advantage of total operational costs over DRP for all examined 15 instances. In particular, given the large cost reduction from battery-related cost savings, although DRPBFS may not always yield the most travelling cost-saving paths, the reduction in total costs is still significant, ranging from 2.91% to 14.34%. In addition, as the problem scale goes up, the cost benefits from fleet sharing are incrementally articulated, with a 2.91% average total cost reduction compared with DRP for 30 nodes (A3-10), 7.02% total cost reduction compared with DRP for 60 nodes (A4-15), and 14.34% total cost reduction compared with DRP for 100 nodes (A5-20). The reasons for this phenomenon are twofold. First, fleet sharing provides much better flexibility for each

flight. Given that each drone does not have to return to its departure depot, the level of *DOD* for each drone can be maintained at a higher level, as opposed to the non-fleet sharing operation, after each flight is completed. Moreover, given how *DOD* affects the *CTF* of a battery, it helps reduce battery-related costs. Second, as the problem size increases (from 30 to 60 to 100 nodes), the drone fleet sharing rate increases significantly (from 3.1% to 7.5% to 15.4% on average). It makes business sense that, as more customers and depots are included, more cost-reduction opportunities emerge. Given that DRPBFS promotes fleet sharing for overall optimisation, the effectiveness of fleet sharing gains more space to have an effect when the problem size is relatively large.

Table 4. Comparison analysis of DRP and DRPBFS

| Ins. | DRP | | | | DRPBFS | | | | | % diff. of batt cost to DRP | % diff. of trans cost to DRP | % diff. of total cost to DRP |
|---------|-----------|-----------|------------|------------|-----------|-----------|---------------|------------|-------------|--------------------------------|---------------------------------|---------------------------------|
| | K | Batt cost | Trans cost | Total cost | K | Batt cost | Trans cost | Total cost | ST rate | | | |
| A3-10-1 | 12 | 9.514 | 5.93 | 15.444 | 13 | 9.345 | 5.979 | 15.324 | 0 | 1.81% | -0.82% | 0.78% |
| A3-10-2 | 13 | 10.025 | 6.359 | 16.384 | 13 | 9.695 | 6.115 | 15.810 | 15.38% | 3.40% | 3.99% | 3.63% |
| A3-10-3 | 20 | 11.202 | 6.854 | 18.056 | 19 | 10.640 | 6.471 | 17.111 | 0 | 5.28% | 5.92% | 5.52% |
| A3-10-4 | 13 | 9.308 | 5.576 | 14.884 | 13 | 9.030 | 5.582 | 14.612 | 0 | 3.08% | -0.11% | 1.86% |
| A3-10-5 | 17 | 9.56 | 5.927 | 15.487 | 16 | 9.277 | 5.794 | 15.071 | 0 | 3.05% | 2.30% | 2.76% |
| Avg. | – | – | – | – | – | – | – | – | 3.1% | 3.32% | 2.26% | 2.91% |
| A4-15-1 | 30 | 17.294 | 10.805 | 28.099 | 28 | 16.554 | 10.405 | 26.959 | 14.29% | 4.47% | 3.84% | 4.23% |
| A4-15-2 | 28 | 16.484 | 10.152 | 26.636 | 28 | 15.452 | 9.568 | 25.020 | 14.29% | 6.68% | 6.10% | 6.46% |
| A4-15-3 | 25 | 15.146 | 9.52 | 24.666 | 23 | 14.200 | 8.940 | 23.140 | 8.70% | 6.66% | 6.49% | 6.59% |
| A4-15-4 | 29 | 19.167 | 11.8 | 30.967 | 27 | 16.811 | 10.544 | 27.355 | 0 | 14.01% | 11.91% | 13.20% |
| A4-15-5 | 31 | 18.973 | 11.432 | 30.405 | 30 | 18.043 | 11.015 | 29.058 | 0 | 5.15% | 3.79% | 4.64% |
| Avg. | – | – | – | – | – | – | – | – | 7.5% | 7.40% | 6.43% | 7.02% |
| A5-20-1 | 50 | 29.217 | 17.88 | 47.097 | 48 | 25.341 | 15.413 | 40.754 | 12.5% | 15.30% | 16.01% | 15.56% |
| A5-20-2 | 45 | 30.073 | 18.811 | 48.884 | 45 | 25.147 | 15.682 | 40.829 | 17.79% | 19.59% | 19.95% | 19.73% |
| A5-20-3 | 45 | 31.069 | 19.59 | 50.659 | 43 | 26.996 | 17.017 | 44.013 | 16.28% | 15.09% | 15.12% | 15.10% |
| A5-20-4 | 42 | 27.864 | 17.621 | 45.485 | 43 | 25.888 | 16.243 | 42.131 | 16.28% | 7.63% | 8.48% | 7.96% |

| | | | | | | | | | | | | |
|---------------|----|--------|--------|-------|----|--------|--------|--------|--------------|---------------|---------------|---------------|
| A5-20-5 | 44 | 30.155 | 18.785 | 48.94 | 43 | 26.637 | 16.548 | 43.185 | 13.95% | 13.21% | 13.52% | 13.33% |
| Avg. | - | - | - | - | - | - | - | - | 15.4% | 14.16% | 14.62% | 14.34% |
| Total avg. | - | - | - | - | - | - | - | - | - | 8.29% | 7.77% | 8.09% |

5.3. Sensitivity analysis

In this section, we conduct sensitivity analysis by varying the form of objective functions, values of battery-related costs and speed of drones. By doing so, we seek to uncover some insights about sustainable drone operation and how it may be affected by the development of technology.

5.3.1. Effect of different objective function

This section discusses the effects of different objective functions with optimising battery costs only (BC), travel costs only (TC) and battery plus travel costs (BC + TC). Accordingly, two more objective functions (Eq. [60] and [61]) are introduced for the pricing sub-problems, which correspond to the BC and TC objectives:

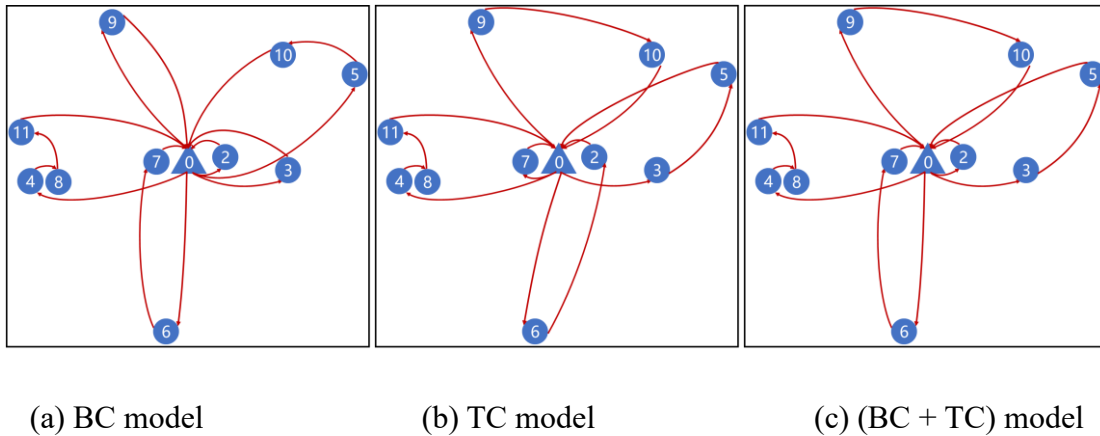
$$\bar{c}_{BC} = z^k - \sum_{(i,j) \in A: i \in O} (\mu_i + v_i) a_{ij}^k - \sum_{i \in C} \sum_{(i,j) \in A} \lambda_i a_{ij}^k + \sum_{(j,i) \in A: i \in O} v_i a_{ji}^k \quad (60)$$

$$\bar{c}_{TC} = \sum_{(i,j) \in A: i \in O} (c_{ij} - \mu_i - v_i) a_{ij}^k + \sum_{i \in C} \sum_{(i,j) \in A} (c_{ij} - \lambda_i) a_{ij}^k + \sum_{(j,i) \in A: i \in O} (c_{ji} + v_i) a_{ji}^k \quad (61)$$

Instance A_2_10_3 generated from the above section is selected for further analysis. Under different objective functions, Table 5 summarises all the paths with their associated costs for the underlying instance. In addition, we draw all the trips surrounding depot 1 under the three models (Fig. 4) to provide a closer presentation of their differences.

Table 5. Trips generated by different objectives for depot 1 in instance A2_10_3

| BC | | TC | | (BC + TC) | |
|-------------------|-----------|--------------------|-----------|------------------|-----------|
| Trips | Batt cost | Trips | Batt cost | Trips | Batt cost |
| [0, 2, 0] | 0.052 | [0, 6, 2, 0] | 0.806 | [0, 2, 0] | 0.052 |
| [0, 3, 0] | 0.462 | [0, 5, 3, 0] | 0.890 | [0, 5, 3, 0] | 0.890 |
| [0, 6, 7, 0] | 0.779 | [0, 7, 0] | 0.048 | [0, 6, 7, 0] | 0.779 |
| [0, 4, 8, 11, 0] | 0.855 | [0, 4, 8, 11, 0] | 0.855 | [0, 4, 8, 11, 0] | 0.855 |
| [0, 9, 0] | 0.684 | [0, 9, 10, 0] | 1.156 | [0, 9, 10, 0] | 1.156 |
| [0, 5, 10, 0] | 0.894 | [1, 13, 21, 1] | 0.554 | [1, 13, 21, 1] | 0.554 |
| [1, 13, 21, 1] | 0.554 | [1, 14, 20, 1] | 0.407 | [1, 14, 20, 1] | 0.407 |
| [1, 14, 20, 1] | 0.407 | [1, 15, 1] | 0.267 | [1, 15, 12, 1] | 0.526 |
| [1, 15, 12, 1] | 0.526 | [1, 18, 12, 1] | 0.612 | [1, 18, 16, 1] | 0.446 |
| [1, 16, 18, 1] | 0.446 | [1, 19, 17, 16, 1] | 1.194 | [1, 19, 17, 1] | 0.999 |
| [1, 19, 17, 1] | 0.999 | | | | |
| Total batt cost: | | 6.658 | | 6.790 | |
| Total travel dis. | | 7,234.8 | | 7,121.6 | |

**Figure 4.** Trips of depot 1 for three types of model

Comparing Fig. 4(a) to the other type graphs, an additional trip is incurred for the BC model. Inevitably, its associated travel cost is higher than the other two, and this also explains the higher total travel distance generated from the BC model compared with the other two in Table 5. However, even with an increased total travel

distance, it is interesting to note that the total battery cost for the BC model is unexpectedly lower than all the others. This leads us to think that the economic effect of a lower *DOD* could be much greater than the additional charging times of a battery. The additional trip in the BC model means that the battery of a drone will be charged one more time than in the other two models; however, with each trip shorter in the BC model, the level of *DOD* will be higher comparatively. If we zoom in to the trip plans for customer 9 across the three models, this further clarifies our finding. Both the TC and BC + TC model choose to service customer 9 together with other customers within one flight, yet the BC model deploys a separate drone deliberately for this customer, so a certain level of *DOD* can be maintained.

5.3.2. Effect of battery disposal cost and purchase cost

This section conducts sensitivity analysis to investigate the effect of the battery disposal cost and purchase cost with the use of instance A2_10_5. The new battery purchase cost starts with our initial setting (i.e., $F = \$500$), yet we exclude the battery disposal cost (i.e., $\delta = 0$). Following this, we analyse both the TC and TC + BC models by decreasing the value of F with step size \$50. Fig. 5 below compares the results in detail.

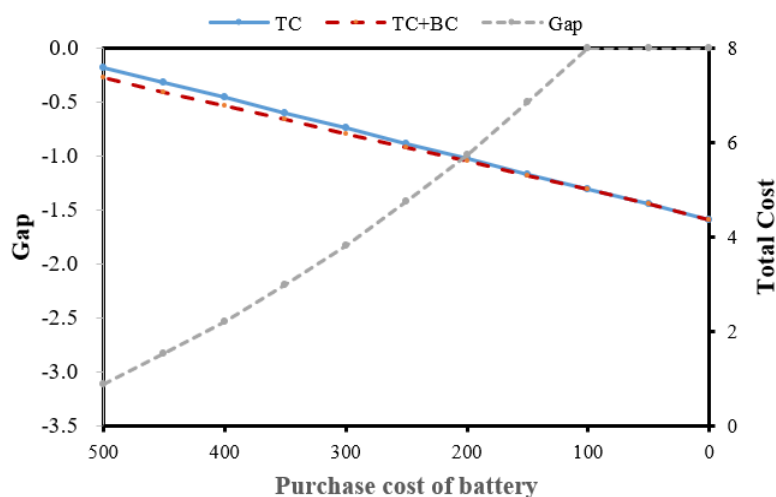
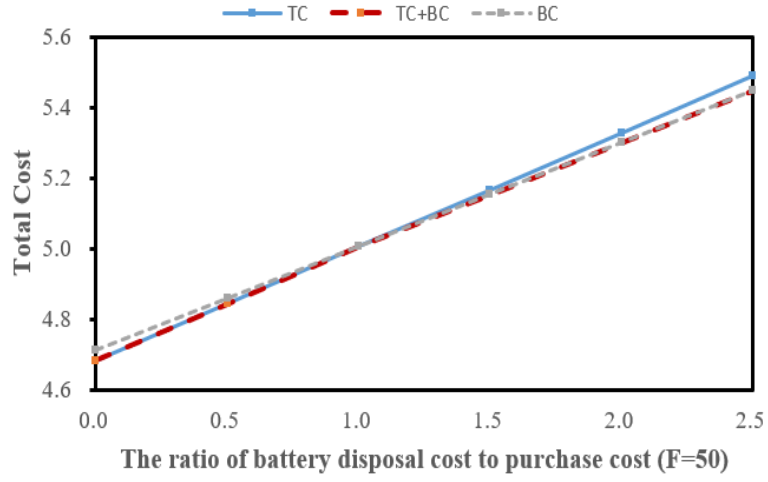


Figure 5. TC versus TC + BC with varying battery purchase cost

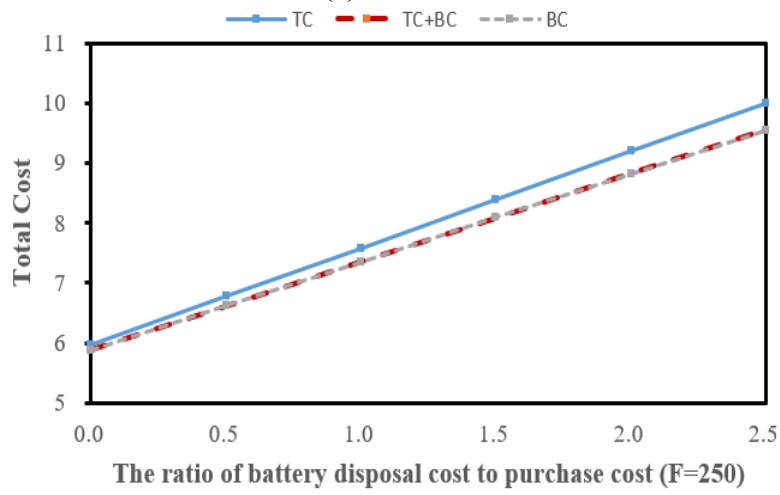
Alongside the decrease in battery purchase cost, the overall optimisation results for TC and TC + BC converge gradually. This means that the shortest path may not necessarily yield an optimal cost control when battery cost is relatively high, as it could accelerate the degradation of batteries. Conversely, if battery purchase costs could be largely reduced, schedules produced from the TC model would be more likely to be the optimal ones and are easier to implement. Therefore, if new battery technology emerges (e.g., reduced battery wear effect or lower manufacturing costs), it would allow drone operators to focus on path planning and simplify their operation.

Next, we evaluate the effects of varying the ratio of battery disposal cost to purchase cost, δ . In addition to the above discussed technology improvement, we believe that battery disposal-related cost could continuously decrease as well. Therefore, we set the ratio δ to 0, 0.5, 1.0, 1.5, 2.0 and 2.5, respectively, to perform experiments under three different battery purchasing costs. Fig. 6 below plots the experimental results in detail.

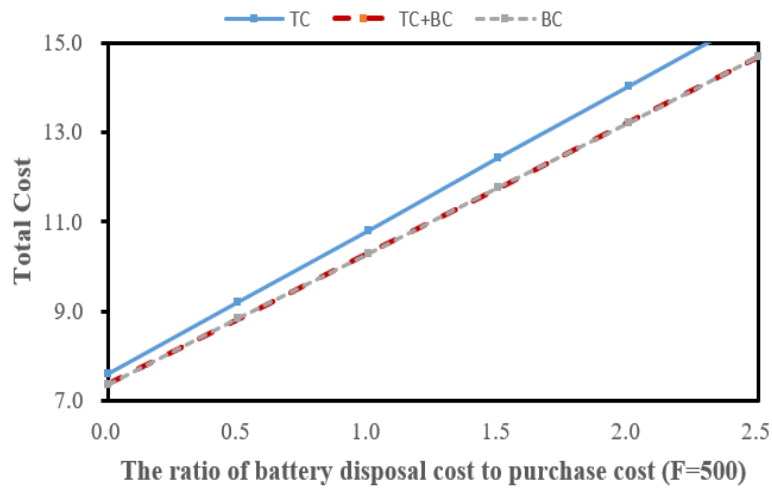
Similar to the experiment results of battery purchase cost, a smaller δ makes TC optimisation closer to TC + BC optimisation for all three battery purchase costs. Notably, when battery purchase cost is \$50 and δ is between 1.0 and 1.3, the TC, TC + BC and BC optimisation is binding, and, if δ goes down further, TC + BC moves towards TC, as the influence of battery-related costs becomes very minor. This means that technological advancements in both battery manufacturing and battery disposal handling will help drone operators better focus on pure distance-based scheduling.



(a) $F = 50$



(b) $F = 250$



(c) $F = 500$

Figure 6. Costs for different δ and F

5.3.3. Effect of drone speed

This section investigates the effect of drone speed on drone fleet sharing operations. Assuming that innovations in drone technology will be interested in developing faster drones, we conduct tests with varying drone speed from 6 to 10 m/s (with incrementing 1 m/s each test) to observe their corresponding changes in total costs and sharing rate. The details of those tests are summarised in Fig. 7 below.

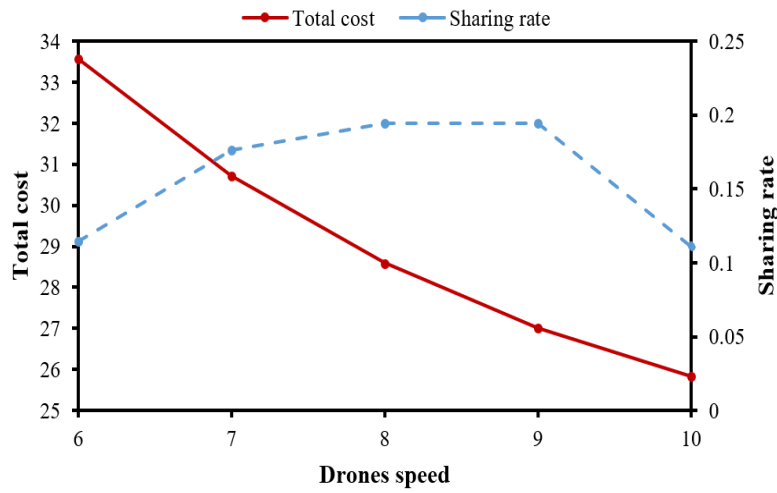


Figure 7. Change in total cost and sharing rate for different drone speeds

Fig. 7 illustrates the change of total costs and fleet sharing rate alongside the increment of drone speed by 1 m/s in each experiment. The plot of the total cost indicates that a continuous reduction can be produced from the improvement of drone speed. It makes sense that the higher speed of drones allows a shorter service lead time. Consequently, if the battery of drones remains at the same capacity, a much higher *DOD* level can be maintained after finishing the same trip. Therefore, a lower battery-related cost can drive the total cost downwards. Examining the change of fleet sharing rate on the other side, it is interesting to see that it increases initially, yet, after a certain speed point (9 m/s), fleet sharing is not favoured any longer, as a sharp decline in rate occurs. The rationale for this is that the increase of drone speed can provide more opportunities and better flexibilities for fleet sharing (e.g., it becomes easier to return a shared drone to its owner); thus, more fleet sharing activities are performed. However, when a drone's speed is fast enough to service multiple times of

customers as only fleet sharing can achieve, fleet sharing starts to lose its economic attractiveness, as the service coverage of one drone is significantly expanded. This implies that the development of drone technology can effectively help extract the value of fleet sharing operation, and eventually replace the role of fleet sharing unless other operational concerns are introduced.

5.3.4. Effect of blockchain

This section varies the sharing ability to represent the use of different fleet sharing protocols. Specifically, thanks to the merits of blockchain technology, the blockchain-enabled platform allows all operators to fully share their demands and drone fleets. However, alternatively, other types of sharing platforms have their sharing rate discounted and range from 10% to 90% (i.e., $\frac{\sum_{i \in O} |C^i|}{|C|}$ ranges from 0.1 to 0.9). Given five different instances with scale of 60 nodes (i.e., four depots and 15 customers with each depot), some key economic factors are examined, including battery cost, travelling cost, total cost, node sharing rate and drone sharing rate. Table 6 summarises the average experimental results.

As the sharing ability of the platform increases (from non-sharing to 100% sharing ability), we see decreases in both costs and total fleet sizes. In particular, platforms with a higher sharing ability contribute significantly to the reduction in both battery-related costs (from \$17.12 to \$16.21 with 5.3% reduction) and total travelling costs (from \$10.63 to \$10.09 with 5.0% reduction). This also means that not only the economic benefits from improved sharing ability but also the overall sustainability is enhanced, as both battery- and travelling-related environment negatives are reduced. In addition, since the total travelling distance is reduced from improved sharing ability, the overall fleet size for all operators can be reduced by up to 3.5% (from 28.2 to 27.2 by average), which can further help reduce capital costs.

Table 6. Experiment results (average) for different types of sharing platforms

| Platform sharing rate | Batt. cost | Travelling cost | Total cost | Node sharing rate | Shared drones | Number of drones | Node serviced by shared resource rate | Drone sharing rate |
|-----------------------|------------|-----------------|------------|-------------------|---------------|------------------|---------------------------------------|--------------------|
| 0% | 17.12 | 10.63 | 27.75 | 0 | 2.8 | 28.2 | 0 | 10.26% |
| 10% | 17.05 | 10.57 | 27.62 | 0.2 | 2.8 | 28.4 | 0.33% | 10.05% |
| 20% | 16.91 | 10.48 | 27.39 | 1 | 2.2 | 27.4 | 1.67% | 8.19% |
| 30% | 16.85 | 10.47 | 27.32 | 1.4 | 2.2 | 27.2 | 2.33% | 8.35% |
| 40% | 16.77 | 10.39 | 27.17 | 3 | 3.6 | 26.6 | 5.00% | 14.16% |
| 50% | 16.65 | 10.33 | 26.98 | 4 | 3.2 | 27 | 6.67% | 12.00% |
| 60% | 16.56 | 10.24 | 26.80 | 5 | 3.2 | 27.2 | 8.33% | 11.89% |
| 70% | 16.48 | 10.21 | 26.69 | 5.8 | 3.2 | 27 | 9.67% | 11.69% |
| 80% | 16.36 | 10.13 | 26.49 | 7 | 3.2 | 26.8 | 11.67% | 11.76% |
| 90% | 16.29 | 10.10 | 26.39 | 7.6 | 3.2 | 27.2 | 12.67% | 11.75% |
| 100% | 16.21 | 10.09 | 26.31 | 7.8 | 2 | 27.2 | 13.00% | 7.45% |

Another interesting finding from this experiment is the change of node sharing rate and drone sharing rate. The node sharing rate refers to how many customer nodes are open to be served by other operators, while the drone sharing rate refers to how many times a drone has different departure and arrival depots. The first helps us understand how often the corresponding sharing platform promotes demand sharing, while the latter helps us understand how often the corresponding sharing platform promotes fleet sharing. The node sharing rate and drone sharing rate are the percentile of these two terms compared with total nodes and total drones, respectively. Figure 8 below plots the change of them alongside the increase of platform sharing ability.

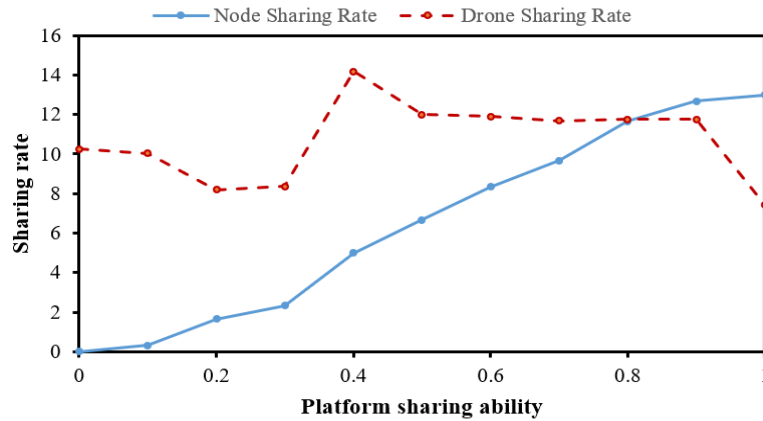


Figure 8. Node sharing rate versus drone sharing rate with respect to different platform sharing ability

As Fig. 8 illustrates, the better sharing ability a platform has, the customer nodes that are open for sharing will have more visits. This makes sense, as the objective of improving platform sharing ability is to allow more customer demands to be served by other operators, if that is economically beneficial. However, counterintuitively, the change of drone sharing rate is not in the same direction as the platform sharing ability change. Instead, it increases initially, as the customer node sharing rate increases; however, after reaching a certain point, it indicates a steady decrease, despite the customer node sharing rate increasing continuously. The reasons for this behaviour are that, when the customer sharing rate level is relatively low, more customer nodes must be served by their corresponding depots. If those unshared customer nodes are far away from their corresponding depot, yet closer to other operators, it will be more cost effective to fly this drone to its nearest depot, instead of returning it to the departure depot. Hence, it increases the chance for drones to be shared. Conversely, for a shared customer node, since it can be served by multiple operators, a nearest depot is more likely to be chosen as the service provider, which makes it more cost effective to return this drone to its departure depot. In this sense, drones are less likely to be shared.

6. Conclusion

In this research, we have studied a DRP based on blockchain-enabled fleet sharing. Specifically, we designed the optimal drone path that makes the best use of drone sharing to minimise both travelling- and battery-related costs. Given the battery wear effect, we proposed a non-linear mixed-integer formulation to describe its associated cost component. We then developed a series of reformulation and solution techniques to make the problem tractable. In comparison with an enumeration-based method and Gurobi-based linear approximation method, our solution showed capability to solve a large-scale test problem within a reasonable computational time.

Experiments were performed to illustrate how the use of fleet sharing affects different operational costs, compared with non-fleet sharing operations. In addition, the metrics of using a blockchain-enabled fleet sharing protocol were demonstrated through comparison with other sharing types. To illustrate the sensitivity of the optimisation to technological evolutions, the effect of varying battery-related costs and drone speed was investigated. In summary, the key findings of this research include the following:

1. Given the strong effects of *DOD* on battery lifespan, it is preferential for drone operators to prioritise an average level of *DOD* in their drone fleet over other decisions.
2. Improving battery-related technologies will help drone operators simplify their overall operational planning, given reduced battery-related effects.
3. A sharing platform with higher sharing ability is beneficial for all operators, as it helps drive down all cost components, yet a higher level of shared demands is not necessarily completed upon increased use of shared drones.
4. To efficiently solve this problem, the exact branch-and-price algorithm was developed. Instances of up to 100 customers can be solved optimally in the allowed time limit by the proposed algorithm, which offers practical applications.

While this paper has contributed new fundamental knowledge to be used in improving both economical and sustainable performance from drone operations, it

should also inspire others to research this new topic, which has been barely studied in the literature to date. This study could be further expanded in two directions. First, our current model does not consider the effects of different payloads. As some research has asserted (e.g., D'Andrea, 2014), the energy consumption rate and flying speed of a drone vary with respect to its payloads. Therefore, it would be interesting to see how this consideration would further affect the optimisation process. Second, our current model assumed that all depots have the same customer distribution. As a result, the demand pattern is relatively balanced and repositions for empty drones are barely required. In reality, an unbalanced demand pattern among different operators is much more common. Therefore, it would be practically meaningful to incorporate empty drone repositioning as part of a DRP with fleet sharing.

7. References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, **52**(4), pp. 965–981.

Amazon, 2020. First prime air delivery. Amazon. Accessed from: <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>.

Asadi, A., Pinkley, S.N., 2021. A stochastic scheduling, allocation, and inventory replenishment problem for battery swap stations. *Transportation Research Part E: Logistics and Transportation Review*, **146**, p. 102212.

Chen, C., Adulyasak, Y., Rousseau, L.M., 2020. Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological*, **139**, pp. 354–387.

Choi, Y., Schonfeld, P.M., 2017. Optimization of multi-package drone deliveries considering battery capacity. *Proceedings of the 96th Annual Meeting of the Transportation Research Board*, Washington, DC (Paper No. 17–05769).

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S.G., Bian, L., 2017. Drones for disaster response and relief operations: a continuous approximation model. *International Journal of Production Economy*, **188**, pp. 167–184.

Cokyasar, T., 2021. Optimization of battery swapping infrastructure for e-commerce drone delivery. *Computer Communications*, **168**, pp. 146–154.

D'Andrea, R., 2014. Guest editorial can drones deliver? *IEEE Transactions on Automation Science and Engineering*, **11**(3), pp. 647–648.

Danna, E., Le Pape, C., 2005. *Branch-and-price heuristics: A case study on the vehicle routing problem with time windows*. Springer, Boston, MA, pp. 99–129.

Darenta, 2021. White Paper. Darenta.io, Accessed from: <https://darenta.io/en.pdf>.

Degraeve, Z., Jans, R., 2007. A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times, *Operations Research*, **55**, pp. 909–920.

Desrochers, M., 1988. *An algorithm for the shortest path problem with resource constraints*. École des hautes études commerciales, Groupe d'études et de recherche en analyse des décisions.

DHL, 2018. DHL parcelcopter. DHL Group. Accessed from: <https://www.dpdhl.com/en/media-relations/specials/dhl-parcelcopter.html>.

Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **47**(1), pp. 70–85.

Dukkanci, O., Kara, B.Y., Bektaş, T., 2021. Minimizing energy and cost in range-limited drone deliveries with speed optimization. *Transportation Research Part C: Emerging Technologies*, **125**, 102985.

Edge, J.S., O'Kane, S., Prosser, R., Kirkaldy, N.D., Patel, A.N., Hales, A., Ghosh, A., Ai, W., Chen, J., Yang, J., Li, S., Pang, M.C., Diaz, L.B., Tomaszewska, A., Marzook, M.W., Radhakrishnan, K.N., Wang, H. Patel, Y., Wu, B., Offer, G.J., 2021. Lithium ion battery degradation: what you need to know. *Physical Chemistry Chemical Physics*, **23**, pp. 8200-8221

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. *Networks*, **44**(3), pp. 216–229.

Figliozzi, M., Tucker, C., Polikakhina, P., 2018. Drone deliveries logistics, efficiency, safety and last mile trade-offs. *Proceedings 7th International Conference on Information Systems, Logistics and Supply Chain, ILS Conference 2018*, July 8–11, Lyon, France.

Fröwis, M., Böhme, R., 2017. In code we trust? In: Garcia-Alfaro, J., Navarro-Arribas, G., Hartenstein, H., Herrera-Joancomartí, J. (Eds.), *Data privacy management, cryptocurrencies and blockchain technology*. Springer International Publishing, Cham, pp. 357–372.

Garey, M.R., Johnson, D.S., 1979. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, San Francisco.

Glaser, F., 2017. Pervasive decentralisation of digital infrastructures: A framework for blockchain enabled system and use case analysis. *In: HICSS 2017 Proceedings*, pp. 1543–1552.

Goodchild, A., Toy, J., 2018. Delivery by drone: an evaluation of unmanned aerial vehicle technology in reducing CO₂ emissions in the delivery service industry. *Transportation Research Part D: Transport and Environment*, **61**, pp. 58–67.

Govindan, K., Jafarian, A., Khdaverdi, R., Devika, K., 2014. Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics*, **152**, pp. 9–28.

Govindan, K., Jafarian, A., Nourbakhsh, V., 2019. Designing a sustainable supply chain network integrated with vehicle routing: a comparison of hybrid swarm intelligence metaheuristics, *Computers & Operations Research*, **110**, pp. 220–235.

Greiner, M., Wang, H., 2015. Trust-free systems—a new research and design direction to handle trust issues in P2P systems: the case of Bitcoin. *In: AMCIS 2015 Proceedings*.

Ha, Q.M., Deville, Y., Pham, Q.D., Ha, M.H., 2018. On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, **86**, pp. 597–621.

Ham, A.M., 2018. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, **91**, pp. 1–14.

Hastig, G.M., Sodhi, M.S., 2020. Blockchain for supply chain traceability: business requirements and critical success factors, *Production and Operations Management*, **29**(4), pp. 935–954.

Hawlitschek, F., Notheisen, B., Teubner, T., 2018. The limits of trust-free systems: a literature review on blockchain technology and trust in the sharing economy. *Electronic Commerce Research and Applications*, **29**, pp. 50–63.

He, L., Ma, G., Qi, W., Wang, X., 2020. Charging an electric vehicle-sharing fleet. *Manufacturing & Service Operations Management*, **23**(2), pp. 267–545.

Hong, I., Kuby, M. Murray, A.T., 2018. A range-restricted recharging station coverage model for drone delivery service planning, *Transportation Research Part C: Emerging Technologies*, **90**, pp. 198–212.

- Hong, Z., Wang, Z., Cai, W., Leung, V.C.M., 2017. Blockchain-empowered fair computational resource sharing system in the D2D network. *Future Internet*, **9**(4), pp. 1-9.
- Hua, S., Zhou, E., Pi, B., Sun, J., Nomura, Y., Kurihara, H., 2018. Apply blockchain technology to electric vehicle battery refueling. *In: Proceedings of the 51st Hawaii International Conference on System Sciences*, pp. 4494–4502.
- Jeong, S., Jang, Y.J., Kum, D., 2015. Economic analysis of the dynamic charging electric vehicle. *IEEE Transactions on Power Electronics*, **30**(11), pp. 6368–6376.
- Jeong, H.Y., Song, B.D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. *International Journal of Production Economics*, **214**, pp. 220-233.
- Kabra, A., Belavina, E., Girotra, K., 2020. Bike-share systems: accessibility and availability. *Management Science*, **66**(9), pp. 3803–3824.
- Katwala, A., 2018. The spiralling environmental cost of our lithium battery addiction. *Wired on Energy*. Accessed from: <https://www.wired.co.uk/article/lithium-batteries-environment-impact>.
- Kirschstein, T., 2020. Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment*, **78**, 102209.
- Lahyani, R., Khemakhem, M., Semet, F., 2015. Rich vehicle routing problems: from a taxonomy to a definition. *European Journal of Operational Research*, **241**(1), pp. 1–14.
- Laporte, G., 2009. Fifty years of vehicle routing. *Transportation Science*, **43**(4), 408–416.
- Lessig, L., 2008. Two Economies: Commercial and Sharing. *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. London: Bloomsbury Academic, pp. 117–176.
- Lu, M., Chen, Z., Shen, S., 2017. Optimizing the profitability and quality of service in carshare systems under demand uncertainty. *Manufacturing & Service Operations Management*, **20**(2), pp. 162–180.
- Lundell, A., Westerlund, J., Westerlund, T., 2009. Some transformation techniques with applications in global optimization. *Journal of Global Optimization*, **43**, pp. 391–405.

- Marinelli, M., Caggiani, L., Ottomanelli, M., Dell’Orco, M., 2017. En-route truck-drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport System*, **12**(4), pp. 253–261.
- Moshref-Javadi, M., Lee, S., Winkenbach, M., 2020. Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E: Logistics and Transportation Review*, **136**, p. 101887.
- Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, **54**, pp. 86–109.
- Nataraj, S., Ferone D., Quintero-Araujo C., Juan A, Festa, P., 2019. Consolidation centers in city logistics: a cooperative approach based on the location routing problem. *International Journal of Industrial Engineering Computations*, **10**(3), pp. 393-404.
- Nesta, 2015. The titans of the sharing economy meet their match. Nesta. Accessed from: <http://www.nesta.org.uk/2016-predictions/titans-sharing-economy-meet-match>.
- Pazaitis, A., De Filippi, P., Kostakis, V., 2017. Blockchain and value systems in the sharing economy: the illustrative case of Backfeed. *Technological Forecasting & Social Change*, **125**, pp. 105–115.
- Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: review and numerical analyses of several models. *Transportation Research Part B: Methodological*, **103**, pp. 158–187.
- Pollard, T., 2020. Electric car batteries: everything you need to know. Carmagazine.co.uk. Accessed from: <https://www.carmagazine.co.uk/electric/ev-car-battery-capacity-tech/>.
- Rapier, R., 2020. Environmental implications of lead-acid and lithium-ion batteries. Forbes. Accessed from: <https://www.forbes.com/sites/rrapier/2020/01/19/environmental-implications-of-lead-acid-and-lithium-ion-batteries/?sh=408cf4b57bf5>.
- Righini, G., Salani, M., 2006. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, **3**(3), pp. 255–273.
- San, K.T., Lee, E.Y., Chang, Y.S., 2016. The delivery assignment solution for swarms of UAVs dealing with multi-dimensional chromosome representation of genetic algorithm. *Proceedings of the Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE Annual, pp. 1–7.

Song, Y., Zhang, J., Liang, Z., Ye, C., 2017. An exact algorithm for the container drayage problem under a separation mode. *Transportation Research Part E: Logistics and Transportation Review*, **106**, pp. 231–254.

Stanciu, A., 2017. Blockchain based distributed control system for edge computing. In: *Proceedings of 2017 21st International Conference on Control Systems and Computer*, pp. 667–671.

Tang, C.S., Veelenturf, L.P., 2019. The strategic role of logistics in the industry 4.0 era. *Transportation Research Part E: Logistics and Transportation Review*, **129**, pp. 1–11.

Teslya, N., Smirnov, A., 2018. Blockchain-based framework for ontology-oriented robots' coalition formation in cyberphysical systems. In: *MATEC Web of Conferences*, **161**.

Troudi, A., Addouche, S.A., Dellagi, S., Mhamedi, A., 2018. Sizing of the drone delivery fleet considering energy autonomy. *Sustainability*, **10**(9), p. 3344.

Ulmer, M.W., Thomas, B.W., 2018. Same-day delivery with a heterogeneous fleet of drones and vehicles. *Networks*, **72**(4), pp. 475–505.

University of Michigan, 2020. Tips for extending the lifetime of lithium-ion batteries. *Michigan News*. Accessed from: <https://news.umich.edu/tips-for-extending-the-lifetime-of-lithium-ion-batteries/>.

Unmanned Cargo, 2017. What do China's delivery drones look like? Unmanned Cargo, Accessed from: <http://unmannedcargo.org/chinese-delivery-drones/>.

Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2013. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, **230**(2), pp. 231–244.

Wang, J., Wu, P., Wang, X., Shou, W., 2017a. The outlook of blockchain technology for construction engineering management, *Frontiers of Engineering Management*, **4**(1), pp. 67–75.

Wang, X., Choi, T.M., Li, S., Shao, S., 2020a. An effective local search algorithm for the multi-depot cumulative capacitated vehicle routing problem. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, **50**(12), pp. 4948–4958.

Wang, X., Choi, T.M., Liu, H., Yue, X., 2016. Novel ant colony optimization methods for simplifying solution construction in vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*, **17**(11), pp. 3132–3141.

Wang, X., Poikonen, S., Golden, B., 2017b. The vehicle routing problem with drones: several worst-case results. *Optimization Letter*, **11**(4), pp. 679–697.

Wang, Y., Li, Q., Guan, X., Fen, J., Liu, Y., Wang, H., 2020b. Collaboration and resource sharing in the multidepot multiperiod vehicle routing problem with pickups and deliveries. *Sustainability*, **12**, p. 5966.

Wang, Z., Sheu, J.B., 2019. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, **122**, pp. 350–364.

Yang, J., Sun, H., 2015. Battery swap station location-routing problem with capacitated electric vehicles. *Computers & Operations Research*, **28**, pp. 217–232.

Yao X., Cheng, Y., Song, M., 2019. Assessment of collaboration in city logistics: from the aspects of profit and CO2 emissions. *International Journal of Logistics Research and Applications*, **22**(6), pp. 576-591.

Yurek, E.E., Ozmutlu, H.C., 2018. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technology*, **91**, pp. 249–262.

Zhang, J., Campbell, J.F., Sweeney II, D.C., Hupman, A.C., 2021. Energy consumption models for delivery drones: a comparison and assessment. *Transportation Research Part D: Transport and Environment*, **90**, p. 102668.

Appendix

A. Nearest neighbour heuristic

Algorithm: Nearest neighbour heuristic

Step 1: First, we generate a path k for the drone starting from the first depot o_1 . Let the path k start from o_1 and end at o_1' (the copy of o_1). Then, execute Step 2 to insert a new customer for the path.

Step 2: If there is only one starting and ending point in the path, find the customer that is the farthest from the starting point and satisfies the constraints—drone capacity, flight duration and time windows. Otherwise, find the qualified customer with the smallest increase in path distance after insertion. Then, go to Step 3.

Step 3: If a qualified customer is found, insert it into the path, and then repeat Step 2. Otherwise, go to Step 4.

Step 4: If all customers have been inserted, the path initialisation procedure is ended; otherwise, a new path will be created with the next depot as the starting and ending point. Go to Step 1 and continue to generate a new path k' for the next depot o_2 .

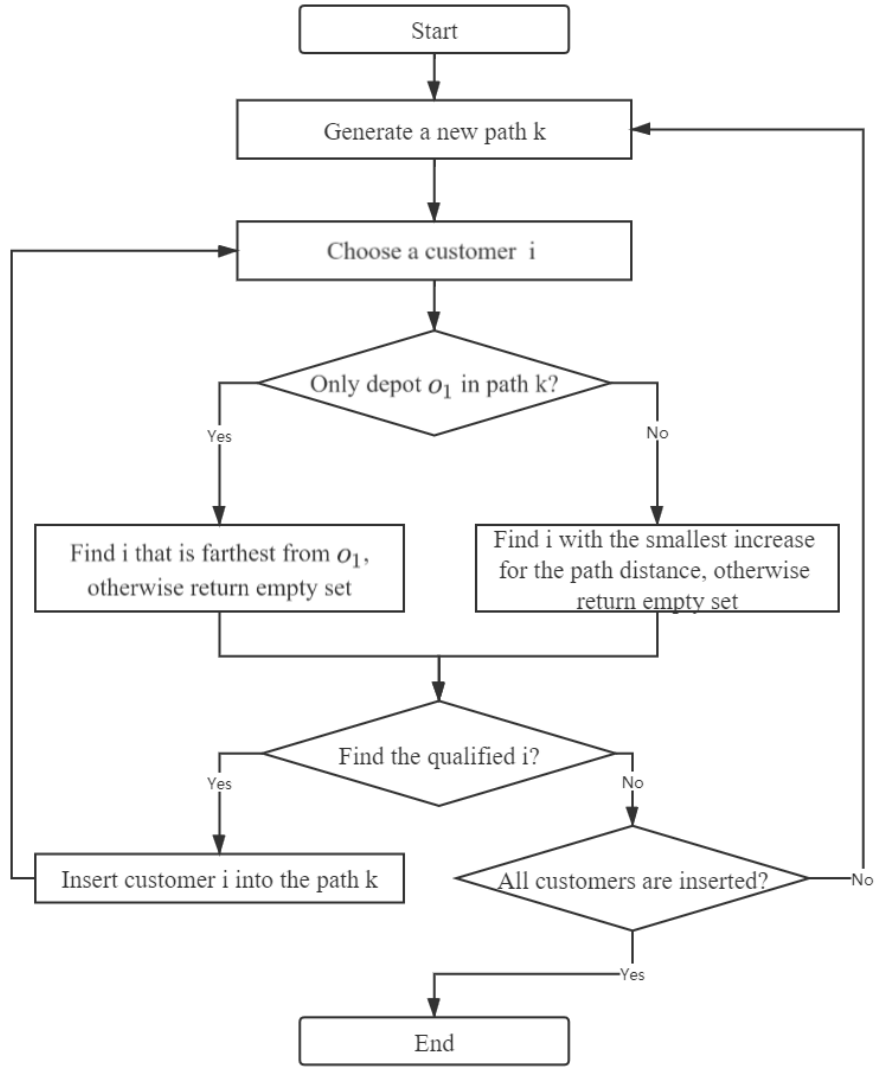


Figure 9. Flowchart for NN heuristic

B. Dual model of master problem

$$\max \sum_{i \in C} \lambda_i + \sum_{i \in O} K_i \mu_i$$

s. t.

$$\sum_{(i,j) \in A: i \in C} a_{ij}^k \lambda_i + \sum_{(i,j) \in A: i \in O} a_{ij}^k (\mu_i + v_i) + \sum_{(i,j) \in A: i \in O} a_{ij}^k v_i \geq z^k + \sum_{(i,j) \in A} a_{ij}^k c_i j, \forall k \in \Omega$$

$$\lambda \leq 0, \mu \geq 0, v \text{ is free}$$

C. Enumeration algorithm

Algorithm 2 Enumerate algorithm

Input: Depot set O , customer set C .

Output: Optimal solution s^*

Initialise: Optimal solution $s^* = \emptyset$, upper bound $UB = \infty$ and solution set $S = \emptyset$

foreach $o \in O$ **do**

 new route $r = (o)$

 new solution $s = \{r\}$

$S = S \cup s$

end for

while $S \neq \emptyset$ **do**

$s = S(1)$ and remove s from S

foreach $i \in O \cup C$ **do**

if i is not visited by s and the customer requirements are satisfied **then**

if $i \in O$ **then**

 add i to the last route of s

 calculate the total cost of s as $cost$

if all customers has been visited by s **then**

if $cost < UB$ and the drone number of each depot is

unchanged **then**

$UB = cost$

$s^* = s$

end if

else

foreach $o \in O$ **do**

$s' = \{s, (o)\}$

$S = S \cup s'$

end for

end if

else

 add i to the last route of s

$S = S \cup s$

end if

end if

end for

end while

D. Piecewise linear approximation with Gurobi (PLA-Gurobi)

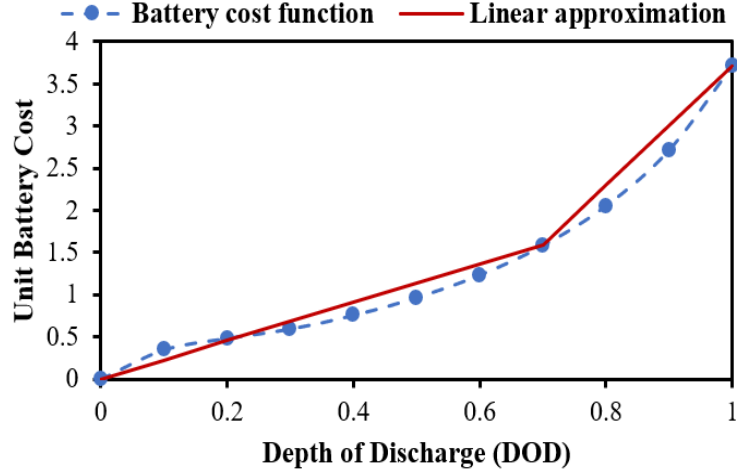


Figure 10. Relationship between battery cost function and its linear approximation

For the non-linear battery cost function, we divide it into two parts, where $x \in [0,0.7]$ and $[0.7,1]$, according to the slopes of the function (shown as Fig. 10). Then the function can be represented as follows:

$$g(DOD) = \frac{(1 + \delta)F}{f_{CTF}(DOD)} = \begin{cases} 2.2654 \cdot DOD, & 0 \leq DOD \leq 0.7 \\ 7.0948 \cdot DOD - 3.3806, & 0.7 < DOD \leq 1 \end{cases} \quad (61)$$

The above linear approximation function can be solved by Gurobi directly.

E. Battery costs results

Table 7. Battery cost sensitivity analysis

| δ | F | TC | | | TC + BC | | | | | |
|----------|-----|-----------|------------|------------|-----------|--------|------------|-------|------------|--------------|
| | | Batt cost | Trans cost | Total cost | Batt cost | Gap | Trans cost | Gap | Total cost | Gap |
| 0.0 | 0 | 0.000 | 4.361 | 4.361 | 0 | 0 | 4.361 | 0 | 4.361 | 0 |
| 0.0 | 50 | 0.322 | 4.361 | 4.683 | 0.322 | 0 | 4.361 | 0 | 4.683 | 0 |
| 0.0 | 100 | 0.644 | 4.361 | 5.005 | 0.644 | 0 | 4.361 | 0 | 5.005 | 0 |
| 0.0 | 150 | 0.967 | 4.361 | 5.328 | 0.8802 | -9.84% | 4.421 | 1.36% | 5.301 | 0.50% |
| 0.0 | 200 | 1.289 | 4.361 | 5.650 | 1.1737 | -9.85% | 4.421 | 1.36% | 5.595 | 0.98% |
| 0.0 | 250 | 1.611 | 4.361 | 5.972 | 1.467 | -9.82% | 4.421 | 1.36% | 5.888 | 1.41% |
| 0.0 | 300 | 1.934 | 4.361 | 6.295 | 1.7607 | -9.85% | 4.421 | 1.36% | 6.182 | 1.80% |
| 0.0 | 350 | 2.257 | 4.361 | 6.618 | 2.0542 | -9.85% | 4.421 | 1.36% | 6.475 | 2.15% |
| 0.0 | 400 | 2.579 | 4.361 | 6.940 | 2.3477 | -9.86% | 4.421 | 1.36% | 6.769 | 2.47% |
| 0.0 | 450 | 2.902 | 4.361 | 7.263 | 2.6413 | -9.85% | 4.421 | 1.36% | 7.062 | 2.76% |
| 0.0 | 500 | 3.224 | 4.361 | 7.585 | 2.9348 | -9.85% | 4.421 | 1.36% | 7.356 | 3.02% |
| 0.5 | 500 | 4.836 | 4.361 | 9.197 | 4.402 | -9.86% | 4.421 | 1.36% | 8.823 | 4.07% |
| 1.0 | 500 | 6.448 | 4.361 | 10.809 | 5.87 | -9.85% | 4.421 | 1.36% | 10.291 | 4.79% |
| 1.5 | 500 | 8.061 | 4.361 | 12.422 | 7.337 | -9.87% | 4.421 | 1.36% | 11.758 | 5.35% |

| | | | | | | | | | | |
|-----|-----|--------|-------|--------|--------|--------|-------|-------|--------|--------------|
| 2.0 | 500 | 9.673 | 4.361 | 14.034 | 8.805 | -9.86% | 4.421 | 1.36% | 13.226 | 5.76% |
| 2.5 | 500 | 11.285 | 4.361 | 15.646 | 10.273 | -9.85% | 4.421 | 1.36% | 14.694 | 6.08% |

F. Drone speed results

Table 8. Drone speed sensitivity analysis

| Speed | Batt cost | Trans cost | Total cost | K | Share time | Share rate |
|-------|-----------|------------|------------|----|------------|------------|
| 6 | 21.042 | 12.523 | 33.565 | 35 | 4 | 11.43% |
| 7 | 18.009 | 12.703 | 30.712 | 34 | 6 | 17.65% |
| 8 | 16.03 | 12.558 | 28.588 | 36 | 7 | 19.44% |
| 9 | 14.491 | 12.522 | 27.013 | 36 | 7 | 19.44% |
| 10 | 13.191 | 12.639 | 25.83 | 36 | 4 | 11.11% |