Computer Communications 168 (2021) 146–154

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Optimization of battery swapping infrastructure for e-commerce drone delivery

Taner Cokyasar

Energy Systems Division, Argonne National Laboratory, 9700 Cass Avenue, Lemont, IL, United States of America

ARTICLE INFO

Keywords:
Drone delivery
Battery swapping for drones
Optimization

ABSTRACT

Drone delivery is widely-researched to alter the current e-commerce delivery convention for providing short delivery lead times. Yet, flight range of drones, constrained by the available battery technology, sets a milestone toward realizing the sole-drone delivery. To tackle with the flight range limitation, locating automated battery swapping machines (ABSM) have been proposed and a few studies modeled the problem. Using the ABSMs, drones can be loaded with fully-charged batteries along the route to a demand location. In this study, we introduce a mixed-integer nonlinear program to model the problem. The objective of the program is to optimally select ABSM locations, determine the delivery-mode choices (drone-only, truck-only, and mixed delivery) of demand locations, find drone delivery routes, and approximate the baseline requirements for the number of drones and batteries needed. The program minimizes the overall delivery system costs including: ABSM, delivery, drone ownership, battery inventory, and service congestion. A cutting-plane method is developed to find exact solutions in finite iterations. Computational experiments showed that the method quickly yields the optimal solution to instances with less than 60 ABSM candidates and 20 demand locations. A case study shows that the optimal drone delivery infrastructure can save almost 20% cost compared to the conventional truck-only delivery. Sensitivity analyses were conducted to reveal the impact of key parameters in the decision-making and found that a decrease in the ABSM and drone costs highly affect the system cost. Code to solve the case study is publicly available at https://gitlab.com/tcokyasar/optimization-of-battery-swapping.

1. Introduction

Drone delivery for e-commerce products has been researched since Amazon, Inc launched the Amazon Prime Air program in 2013 [1]. Not long after, German parcel delivery company, DHL, also began testing the drone delivery in 2013 [2]. Google [3], Fedex [4], JD.com [5], and many other e-commerce and postal service companies formed research teams to investigate the feasibility of the drone delivery. On the other hand, the U.S. government, through the Federal Aviation Administration (FAA), limited drone operations under FAA Part 107 but granted waivers to applicants, and created a data exchange platform called LAANC to integrate drone operations with the current airspace system [6].

Amazon filed a patent illustrating a bee-hive-shaped fulfillment center (FC) design to operate drone deliveries [7]. This patent inspired researchers to suggest three main alternative drone delivery networks: (1) Installing charging or battery swapping stations, so called automated battery swapping machines (ABSMs) [8–12], (2) using delivery trucks in-tandem with drones [13–19], and (3) using public transport to rest drones during delivery [20–22]. Most of these alternatives were proposed to eliminate/relax the flight range limitation of drones stemming from the current state of the battery technology that does not support economically viable long-distance travels. Hong et al. [10] addressed the expensive investment cost of Amazon's FC design and proposed locating commercially available ABSMs [23] to reduce the infrastructure cost and eliminate the flight range limitation. They described a delivery network in which batteries of drones are replaced by ABSMs to allow drones reach farther distances.

In this manuscript, we propose a model to solve the e-commerce drone delivery (E-CDD) problem. The E-CDD problem requires strategically locating ABSMs, determining the approximate number of drones an batteries to be possessed, and partitioning the demand of a given set of locations into drone-only, truck-only, and mixed delivery modes. To solve the problem, the long-term investment and operational costs are minimized. We specifically emphasize the word *only* as we do not consider a drone and truck in-tandem delivery. Instead, each order is delivered with a single channel (drones or trucks) and a demand location can be served by both delivery modes for different orders, which is called a mixed delivery. To model the problem, we develop a mixed-integer nonlinear program (MINLP) that locates ABSMs and routes drones while considering a possible congestion at battery swapping operations through the queueing theory. The MINLP is converted

E-mail address: tcokyasar@anl.gov.

https://doi.org/10.1016/j.comcom.2020.12.015
Received 12 July 2020; Received in revised form 18 August 2020; Accepted 14 December 2020
Available online 28 January 2021

to a mixed-integer quadratically-constrained program (MIQCP) by relaxing the queueing constraints to solve the problem via commercially available solvers. The relaxed constraint is iteratively introduced to the program based on a cutting-plane method [24]. The method provides an exact solution to the problem in finite number of iterations.

The contributions of the study are listed as follows.

- We differently model the E-CDD problem by considering the battery inventory and queueing impact as variable. Hence, the program optimally determines the number of batteries to be stocked and finds the optimal number of drones to be queued for receiving battery swapping service at each swapping location.
- A comprehensive literature review with objective comparisons is provided.
- We provide an exact solution method that can be implemented to similar programs. Further, we test the performance of this method and compare it to an alternative modeling approach.
- We develop a case study by implementing the model in a portion of the Chicago metropolitan area and publicly share the data structure and the code. We also provide sensitivity analyses to address the future scenarios as the drone technology is still emerging.

2. Literature review

The E-CDD literature is centered around three aforementioned approaches. Since this study is related to the first approach (installing charging or battery swapping stations), we mostly focus on the relevant literature and briefly review other approaches.

To the best of our knowledge, Hong et al. [10] first proposed to locate ABSMs for flight range extensions. The authors located the machines while maximizing the demand coverage and solely focusing on the drone delivery. In our model, we minimize the long-term costs and introduce trucks as an economically competitor delivery mode-choice. Furthermore, the queueing intensity at ABSM locations are penalized to control the delivery lead time and evenly distribute drones to different routes to avoid congestion at ABSMs. Huang and Savkin [11] developed a similar modeling approach to [10] but found the minimum number of ABSM deployment rather than deploying ABSMs to a given set of locations. They did not consider the congestion possibility at ABSMs and used simulation to solve the problem.

Cokyasar et al. [8] proposed a similar modeling approach to this study but did not consider optimizing the numbers of drones and batteries. Cokyasar et al. [9] extended the previous work by optimizing the bi-directional routing of drones. Comparatively, this study relaxes the routing optimization to one-way – from the FC to the demand location – but introduces mixed delivery to each location and optimizes the battery inventory. Shavarani et al. [12] focused on locating both multiple FCs and ABSMs using fuzzy variables to represent uncertainties in demand and travel time. They considered the congestion impact through an $M/G/k$ queue yet assumed a constant queueing time and did not model for the battery inventory. Moreover, they proposed a metaheuristic solution approach which does not guarantee global optimality. Hassija et al. [25] studied a similar problem of scheduling drone visits to charging locations using a game theoretic approach. They provided a useful pictorial representation of the designed network, which is similar to our network layout. They used simulation to solve the problem.

Table 1 provides an objective comparison of this study with the closely relevant literature. In each article, model type, objective, and features are investigated. The common model types are mixed-integer linear programming (MILP) and MINLP. The model objectives are centered around minimizing the overall system cost and maximizing the service coverage area. These studies were evaluated based on six features: (1) Extent of drone routing (whether one-way or round-trip), (2) number of FCs considered (single or multiple), (3) availability of alternative delivery mode-choices, (4) consideration of service congestion through a queueing model, (5) internalization of the number of drones in the decision-making process, and (6) integration of the battery inventory requirements in the modeling.

Apart from the aforementioned studies, there is a large body of literature focused on different applications of drones, from security surveillance to disaster aid. Al-Turjman et al. [26] optimized the deployment of drones over a network to monitor static and mobile targets. The authors aimed to minimize the number of drones utilized and to maximize the coverage area. Similarly, Khan et al. [27] studied a drone monitoring design to prevent traffic accidents. Rabta et al. [28] developed an MILP to solve the last-mile drone distribution in disaster relief operations. They assumed that recharging stations are readily built on the network and their location decision was not considered.

In our study, we assume the drone paths between any two locations are known. Aggarwal and Kumar [29] provided a comprehensive review on how paths can be found. Bacanli and Turgut [30] considered using drones to deliver packets and messages in opportunistic networks via drone aid and assessed the message delay distributions.

3. Methodology

In this section, we formally describe the problem, present the model structure and formulations, and propose an alternative solution approach to benchmark the original model.

3.1. Problem description

We can describe the E-CDD problem as follows. An e-commerce company that currently uses trucks for delivery aims to quickly deliver a portion of their parcels using drones. Yet, the customers are spatially distributed and the current battery technology does not allow the delivery drones to serve all customers within the service range of an FC. Therefore, the company plans to locate ABSMs and to partition the demand of an FC into trucks and drones. The E-CDD problem is to find these optimal locations through drone routing and determine the ramifications of the delivery mode-choice for each demand location, while minimizing long-term investment and operational costs, including machine, delivery, drone ownership, battery inventory, and delivery lead time (equivalently, customer satisfaction through delivery speed) costs. We list the following assumptions that will play important roles in our model settings. Note that we use the terms *demand location* and *demand vertex* interchangeably.

- The demand is assumed to follow a Poisson process, one unit is demanded at a time, and each drone can deliver only one unit in a single delivery because of the currently available delivery drone designs. The Poisson process in e-commerce demands is commonly-adopted in the literature due to the infrequency of orders at vertex-level [12,31,32].
- ABSMs across different locations are identical and battery swapping time is constant.
- Due to a possible congestion during the battery swapping operation, we assume a queue is formed to accommodate drones waiting for the swap service, and an adequate queueing space at each candidate location is available.
- Each ABSM location can house at most one swapping machine, and these locations do not carry product inventory. We make this assumption because one of the targets of constructing a battery swapping infrastructure is to lower the long-term investment costs by eliminating the need for FCs at many locations. However, drones need to traverse the delivery route after each delivery.
- Drones are assumed to be identical and to fly at a constant speed.
- Drones are equipped with fully-charged batteries at the FC and ABSM locations.
- Drones are operated under the FAA regulations and waivers for critical rules (such as visual line-of-sight only) that may preclude automated drone deliveries are assumed to be granted.

Table 1
Systematic literature review of drone delivery with battery swapping.

Article	Model type	Model objective	Model features
Cokyasar et al. [8]	MINLP	Minimize ABSM, drone delivery, truck delivery, and congestion costs	(1) One-way drone routing, (2) Single FC, (3) Competition with trucks, (4) Variable service congestion, (5) No number of drones required, (6) No number of batteries required.
Cokyasar et al. [9]	MINLP	Minimize ABSM, drone delivery, truck delivery, drone ownership, and congestion costs	(1) Round-trip drone routing, (2) Single FC, (3) Competition with trucks, (4) Variable service congestion, (5) Number of drones required, (6) No number of batteries required.
Hong et al. [10]	MILP	Maximize the demand covered by drones	(1) One-way drone routing, (2) Single FC, (3) No competition, (4) No congestion, (5) No number of drones required, (6) No number of batteries required.
Huang and Savkin [11]	Simulation	Minimize the number of charging or battery swapping locations while satisfying a coverage ratio and connectivity requirements	(1) One-way drone routing, (2) Single FC, (3) No competition, (4) No congestion, (5) No number of drones required, (6) No number of batteries required.
Shavarani et al. [12]	MINLP	Minimize FC, ABSM, drone ownership, and drone delivery costs	(1) One-way drone routing, (2) Multiple FCs, (3) No competition, (4) Fixed service congestion, but flexible pick-up time, (5) Number of drones required, (6) No number of batteries required.
This study	MINLP	Minimize ABSM, drone delivery, truck delivery, drone ownership, battery inventory, and congestion costs	(1) One-way drone routing, (2) Single FC, (3) Competition with trucks and mixed delivery, (4) Variable service congestion, (5) Number of drones required, (6) No number of batteries required.

- Maximum travel range for drones is assumed to be constant regardless of the weather conditions, wind speed and direction, and the loaded parcel weight.
- Drones follow the same route in both directions of a delivery flight. A route connects the FC, ABSM locations, and the customer. To form possible drone delivery routes, our model decides where to locate ABSMs out of a given set of candidate locations. Since a long-term solution is derived, the impact of bi-directional routing on the overall system cost can be ignored. The problem is solved on a static-basis, in which information are available and additional demand does not emerge during the planning horizon. In a dynamic operational-level drone routing, however, our assumption could have led to a non-optimal system design because the routing could be affected by the incoming delivery demands.
- Truck routes are assumed to be predefined and their routing is not studied. Further, we assume that shifting delivery service from trucks – the current mode-choice – to drones will not impact the truck delivery routes or costs. In the development phase, drones may not be able to replace trucks for majority of deliveries; thus, the slight change in the truck routing may be ignored in a strategic decision-making framework.
- An ABSM may pause providing service due to an unavailability in the fully-charged battery inventory. In the queueing theory, this phenomenon is described as the *server vacation* which will be elaborated later in the section.

In the optimized delivery network, a drone, loaded with a fully-charged battery and the parcel at the FC, travels to the demand location by hopping from one ABSM to another, where its battery is replaced with a fully-charged one. Each flight from a vertex to another is called a *hop*. The drone may need to wait in a queue to receive an automated battery swapping service. This queue is a natural outcome of the stochastic demand because many orders can be placed simultaneously and drones may request battery swapping service at an ABSM in-tandem. In the case of a fully-charged battery unavailability, the drone can be subject to longer wait times. Once the drone arrives at the demand location and delivers the parcel, it returns back to the FC using the same route and receiving battery swapping service at the same locations in the reverse sequence. If a demand location is served by a truck, the delivery is traditionally carried out. Delivery to a demand location can be made by both drones and trucks by partitioning the location's total demand into the delivery mode-choices.

3.2. Model structure and formulation

To begin with notations, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph representation of the problem, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of feasible edges drones can traverse. The graph $\mathcal{G}$ spans the service network of a given FC denoted by the set element $\{0\}$ and includes a set of demand locations denoted by $\mathcal{D}$ and a set of candidate ABSM locations denoted by $\mathcal{J}$, where $\mathcal{V} = \{0\} \cup \mathcal{J} \cup \mathcal{D}$. In the set of edges, $\mathcal{E}$, we define feasible flights between origins and destinations, where the feasibility is bounded by the maximum time a drone can fly without a battery swapping. According to this definition, a drone can hop from the FC or an ABSM location to another ABSM location or the customer vertex, though every delivery begins at the FC. The notations used in the model are provided in Table 2 and follow similar conventions of [9]. To ease reading, note that we use calligraphic letters to represent sets, uppercase Roman letters for parameters (e.g., T) with an exception of L_j, lowercase Roman letters for variables (e.g., x_j), and lowercase Greek letters (e.g., ϕ for fixed, δ for drone delivery, and τ for trucks) -except for λ and μ- as superscripts to modify parameters. The notations, λ, μ, and L, are commonly associated with the queueing theory. Hence, we follow the literature and exempt these notations from our conventions.

At ABSM locations, batteries are swapped following a first-come, first-served (FCFS) discipline. To model the congestion impact at ABSMs and consider the battery inventory, we use a special type of $M/D/1$ queueing model that is called "$M/D/1$ queue with exponential server vacations" [33]. In this queueing model, the demand arrival follows a Poisson process, service time is deterministic (i.e., fixed), one server works in an FCFS basis, and the server goes to a vacation (time of which follows an exponential distribution) after each service with a probability P following a Bernoulli schedule. In our problem layout, we interpret this queueing model as follows. Each order made by a demand location is equivalent to a drone arrival at an ABSM location, if the order is delivered by drones. Since the demand (λ_d^κ) follows a Poisson process, the arrival of drones demanding a battery swapping service (λ_j) also follows the same process. An ABSM is called the server, only one of them can be housed at each candidate location $j \in \mathcal{J}$, and they can automatically swap and recharge batteries. The battery

Table 2
Set, parameter, and variable definitions.

Set	*Definition*
$\mathcal{V}$	Set of vertices, indexed by $i \in \mathcal{V} = \{0\} \cup \mathcal{I} \cup \mathcal{D}$.
$\mathcal{D}$	Subset of vertices for demand locations, indexed by $d \in \mathcal{D}$ and $\mathcal{D} \subset \mathcal{V}$.
$\mathcal{I}$	Subset of vertices for battery swapping machine candidate locations, indexed by $i \in \mathcal{I}$ and $\mathcal{I} \subset \mathcal{V}$.
$\mathcal{E}$	Set of edges, indexed by the tuple $(i,j) \in \mathcal{E} = \{(i,j) \mid i \in \mathcal{V} \setminus \mathcal{D} \wedge j \in \mathcal{V} \setminus \{0\} \wedge i \neq j \wedge T_{ij} \leq R\}$.
$\mathcal{I}_j(\{\cdot\})$	Subset of feasible origin vertices of a given vertex $j \in \mathcal{I}$ that can be traveled from without battery swapping, $\mathcal{I}_j(\{\cdot\}) = \{i \mid i \in \{\cdot\} \wedge T_{ij} \leq R \wedge i \neq j\}$, where $\{\cdot\}$ represents the input set for index i.
Parameter	*Definition*
C_j^{ϕ}	Fixed cost of locating a battery swapping machine at location $j \in \mathcal{I}$ per time unit.
C^{δ}	Unit drone transportation cost per time unit.
C^{τ}	Unit truck transportation cost per distance.
C^{θ}	Unit drone ownership cost per time unit.
C^{β}	Unit battery cost per time unit.
D_d	Distance to be traveled to vertex $d \in \mathcal{D}$ by trucks to deliver packages.
P	Probability of confronting swap service interruption due to fully-charged battery inadequacy.
R	Maximum travel time without battery swapping.
S	Stability coefficient.
T	Time required to fully charge an empty battery.
T_{ij}	Travel time for drones on any origin–destination pair (i,j), where $i,j \in \mathcal{V}$.
λ_d^{κ}	Demand rate at vertex $d \in \mathcal{D}$ during the planning horizon.
μ	Service rate of battery swapping machine during the planning horizon.
Variable	*Definition*
b_j	Minimum number of batteries needed at the ABSM location $j \in \mathcal{I}$.
L_j	Average number of drones receiving battery swap service at vertex $j \in \mathcal{I}$.
r_d	Ratio of the demand at vertex $d \in \mathcal{D}$ delivered by drones.
w	Average number of drones used.
x_j	$\begin{cases} 1, & \text{if a battery swapping machine is located at vertex } j \in \mathcal{I}, \\ 0, & \text{otherwise.} \end{cases}$
y_{ijd}	$\begin{cases} 1, & \text{if any portion of demand at vertex } d \in \mathcal{D} \text{ is served by drones traversing edge } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$
z_d	Ratio of the demand at vertex $d \in \mathcal{D}$ delivered by trucks.
λ_j	Demand rate at vertex $j \in \mathcal{I}$.

swapping operation takes a fixed time that is called the service time, μ^{-1}. We assume each battery can be fully-charged in T time units regardless of its state-of-charge at arrival. Eq. (1), borrowed from [33], calculates the average number of drones receiving a battery swapping service at vertex $j \in \mathcal{I}$, denoted by L_j, under the given demand arrival rate, λ_j. In [33], the equation was written using different notations (i.e., b instead of μ^{-1}, β instead of T, and p instead of P), and a function defining the probability of the server being idle was embedded in the equation. Hence, we re-write the equation by expanding this function and using our notation system. This metric is represented by the function $f(\lambda_j)$. Using this queueing performance metric, our model will penalize the objective function because high number of drones at ABSM locations retard the delivery service, cause customer dissatisfaction, and can even lead to a total delivery system failure due to too high congestion. Penalizing with L_j helps better distributing drones into different routes and mitigates the congestion.

$$L_j = f(\lambda_j) = \frac{\frac{\lambda_j^2}{\mu^2} + \frac{2P\lambda_j^2}{T\mu} + \frac{2P\lambda_j^2}{T^2}}{2\left(1 - \frac{\lambda_j}{\mu} - \frac{P\lambda_j}{T}\right)} + \frac{\lambda_j\left(\mu P + \mu T P + T^2 + T P\right)}{T^2 \mu \left(1 + \frac{P}{T}\right)} \qquad \forall j \in \mathcal{I}, \tag{1}$$

We use such a queueing model, rather than the classic $M/D/1$, to account for the cost of batteries. Although batteries individually do not cost a lot, keeping a high number of stocks can impact the optimal delivery mode-choice decisions. Furthermore, the life-cycle for batteries is often short due to degradation. Shavarani et al. [12] estimates that the life-cycle of batteries is two years, each may cost \$200, and each ABSM location may keep 200 units of batteries. Therefore, using the model in [33] coupled with an approximation in [34], we estimate the minimum number of batteries needed under a given probability P of fully-charged battery unavailability and the battery charging time T. Without loss of generality, we present the following MINLP to model the problem.

$$\text{Min } \mathbf{C} = \sum_{j\in\mathcal{I}} C_j^{\phi} x_j + \sum_{\substack{(i,j)\in\mathcal{E},\\ d\in\mathcal{D}}} 2\lambda_d^{\kappa} C^{\delta} T_{ij} y_{ijd} r_d + \sum_{d\in\mathcal{D}} \lambda_d^{\kappa} C^{\tau} D_d z_d + C^{\theta} w$$
$$+ \sum_{j\in\mathcal{I}} C^{\beta} b_j + \sum_{j\in\mathcal{I}} C^{\delta} f(\lambda_j), \tag{2}$$

subject to,

$$r_d + z_d = 1 \qquad \forall d \in \mathcal{D}, \tag{3}$$

$$r_d \leq \sum_{i\in\{i \mid i\in\mathcal{I} \wedge T_{id} \leq R\}} y_{idd} \leq \mathbb{M}_1 r_d \qquad \forall d \in \mathcal{D}, \tag{4}$$

$$2T_{id} y_{idd} \leq R \qquad \forall i \in \{i \mid i \in \mathcal{I} \wedge T_{id} \leq R\}, d \in \mathcal{D}, \tag{5}$$

$$\sum_{\mathcal{I}_j(\mathcal{V}\setminus\mathcal{D})} y_{ijd} = \sum_{i\in\mathcal{I}_j(\mathcal{V}\setminus\{0\})} y_{jid} \qquad \forall j \in \mathcal{I}, d \in \mathcal{D}, \tag{6}$$

$$\sum_{\substack{i\in\mathcal{I}_j(\mathcal{V}\setminus\mathcal{D}),\\ d\in\mathcal{D}}} y_{ijd} \leq \mathbb{M}_2 x_j \qquad \forall j \in \mathcal{I}, \tag{7}$$

$$\lambda_j = \sum_{\substack{i\in\mathcal{I}_j(\mathcal{V}\setminus\mathcal{D}),\\ d\in\mathcal{D}}} 2\lambda_d^{\kappa} y_{ijd} r_d \qquad \forall j \in \mathcal{I}, \tag{8}$$

$$\lambda_j \leq \frac{ST\mu}{T + P\mu} \qquad \forall j \in \mathcal{I}, \tag{9}$$

$$w \geq \sum_{\substack{(i,j)\in\mathcal{E},\\ j\notin\mathcal{D}}} T_{ij}\lambda_j + \sum_{j\in\mathcal{I}} f(\lambda_j), \tag{10}$$

$$b_j \geq T\lambda_j + \Phi^{-1}(1 - P)\sqrt{T\lambda_j} \qquad \forall j \in \mathcal{I}, \tag{11}$$

$$\begin{aligned} &x_j \in \{0,1\},\ y_{ijd} \in \{0,1\},\ z_d \in [0,1],\ r_d \in [0,1], \\ &z_d,\ r_d,\ \lambda_j,\ L_j,\ w,\ b_j \in \mathbb{R}^{+}. \end{aligned} \tag{12}$$

In each summation of objective function (2), respectively, ABSM, drone delivery (per travel time), truck delivery (per distance), drone ownership, battery inventory, and the queueing penalty costs are accounted. Note that all the cost components are in the units of cost over time, where the time is the planning horizon, e.g., a day. The function, $f(\lambda_j)$, in (2) is the average number of drones receiving a battery swapping service at vertex $j \in \mathcal{I}$ based on a given demand rate—λ_j—and other constants: P, T, and μ. The constraint set (3) ensures that the demand at vertex $d \in \mathcal{D}$ is completely met/delivered by drones and trucks. The constraint set (4) enforces the routing to demand vertex $d \in \mathcal{D}$ for drones in case a portion of the demand is delivered by drones, i.e., $\sum_{i\in\{i|i\in\mathcal{I}\wedge T_{id}\le R\}} y_{idd} = 1$, if $r_d > 0$ and vice versa. The constant, $\mathbb{M}_1$ should be large enough to allow r_d take small values but reasonable enough that would not lead to numerical errors, e.g., $\mathbb{M}_1 = 100$ can be considered a reasonable choice allowing $\min(r_d) = 0.01$. The constraint set (5) satisfies that the total travel time from and to the immediate ABSM of $d \in \mathcal{D}$ to vertex $d \in \mathcal{D}$ does not exceed the maximum travel time without battery swapping. We use the constant two because we assume the travel time on both directions are symmetric. One, rightfully arguing that drones may fly longer distances (without a battery swapping) after the parcel is delivered due to the drop in the weight, can adjust this constant accordingly. The constraint set (6) assures that a drone receiving a battery swapping service at $j \in \mathcal{I}$ leaves the vertex after the service is completed, a.k.a. node-balance constraint. The constraint set (7) ensures that a delivery route to vertex $d \in \mathcal{D}$ may include edge (i, j) only if an ABSM is located at vertex $j \in \mathcal{I}$. The constant, $\mathbb{M}_2$, in this constraint set should satisfy $\mathbb{M}_2 \ge |\mathcal{D}| \max_{j\in\mathcal{I}}(|\mathcal{I}_j(\mathcal{V} \setminus \mathcal{D})|)$.

The constraint set (8) defines the demand rate at vertex $j \in \mathcal{I}$ in terms of the routing variables (y_{ijd}) and the ratio of drone-delivered demand flowing through the vertex. The constant two in this constraint exists because drones are assumed to stop by the same location $j \in \mathcal{I}$ in return. Once again, this constant can be adjusted to consider different situations. The constraint set (9) enforces the stability conditions under which the steady state exists [33]. The original form of the condition is $\lambda_j < \frac{T\mu}{T+P\mu}$; yet, we use S as a stability coefficient to relax the strictness. Constraint (10) approximates the system-wide average number of drones used by accounting for the drones traveling on the edges and the ones at ABSM locations.

The constraint set (11) is borrowed from [34] and approximates the minimum number of battery inventory to hold to guarantee that $1 - P$ percent of the drones will receive batteries that have been recharged for T hours or more. The function $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of the standard Normal, approximates the Poisson demand, and is treated as a constant input given the parameter P. Mak et al. [34] uses this approximation for an electric vehicle battery swapping problem, which is quite similar to our problem layout although they do not consider a queueing perspective. The authors state that such an approximation is commonly-used in inventory control and accurate when $b_j \ge 10$. We especially used this approximation coupled with the queueing model of [33] to overcome the challenge in accounting the battery costs. The probabilistic approaches in the two models complement each other and help us quantitatively assess the importance of battery inventory within the strategic decision-making framework. Lastly, The constraint set (12) indicates the domains of variables, and all parameters are in $\mathbb{R}^+$.

The model is nonlinear due to $y_{ijd}r_d$, L_j, and b_j. The following sub-section will address how to deal with these nonlinear terms.

3.3. An exact solution method

We begin with linearizing the term $y_{ijd}r_d$ in (2) since this is quite straightforward. We introduce a set of continuous variables $g_{ijd} = y_{ijd}r_d$, where $g_{ijd} \in [0, 1]$ $\forall (i, j) \in \mathcal{E}$, $d \in \mathcal{D}$. We replace (2) and (8) with (13) and (14), respectively. The constraint set (15) is introduced to enforce g_{ijd} to represent the multiplication.

$$\text{Min } \mathbf{C} = \sum_{j\in\mathcal{I}} C_j^{\phi} x_j + \sum_{\substack{(i,j)\in\mathcal{E},\\ d\in\mathcal{D}}} 2\lambda_d^{\kappa} C^{\delta} T_{ij} g_{ijd} + \sum_{d\in\mathcal{D}} \lambda_d^{\kappa} C^{\tau} D_d z_d + C^{\theta} w$$
$$+ \sum_{j\in\mathcal{I}} C^{\beta} b_j + \sum_{j\in\mathcal{I}} C^{\delta} L_j, \tag{13}$$

$$\lambda_j = \sum_{\substack{i\in\mathcal{I}_j(\mathcal{V}\setminus\mathcal{D}),\\ d\in\mathcal{D}}} 2\lambda_d^{\kappa} g_{ijd} \qquad \forall j \in \mathcal{I}, \tag{14}$$

$$g_{ijd} \le y_{ijd}, \; r_d + y_{ijd} - 1 \le g_{ijd} \le r_d \qquad \forall (i, j) \in \mathcal{E}, d \in \mathcal{D}. \tag{15}$$

Next, we define $e_j = \sqrt{\lambda_j}$ and replace (11) with (16). Since commercially available solvers can deal with non-convex quadratic constraints, we introduce (17) to enforce $e_j = \sqrt{\lambda_j}$.

$$b_j \ge T\lambda_j + \Phi^{-1}(1 - P)\sqrt{T} e_j \qquad \forall j \in \mathcal{I}, \tag{16}$$

$$e_j^2 = \lambda_j \qquad \forall j \in \mathcal{I}, \tag{17}$$

Finally, a strict nonlinearity remains in the definition of L_j in Eq. (1). To deal with this, we solve the MIQCP formed by (13), (3)–(7), (9), (10), and (12)–(17) (which will be henceforth called the *initial program*). Then, we iteratively introduce cutting-plane constraints explained in [24]. To do this, we need to ensure that (1) is convex (because of the minimization in the objective function) in λ_j in the domain we solve the problem. Theorem 1 states the convexity conditions and its proof is provided.

Theorem 1. *The function $f(\lambda_j)$ is convex in λ_j in the domains: $T, \mu, P, \lambda_j \in \mathbb{R}^+$ satisfying $\lambda_j < \frac{T\mu}{T+P\mu}$.*

Proof. To prove Theorem 1, we take the second order derivative of $f(\lambda_j)$ with respect to λ_j. If $\frac{d^2 f(\lambda_j)}{d\lambda_j^2} \ge 0$ is satisfied, we can prove $f(\lambda_j)$ is convex in the aforementioned domain. The second order derivative is

$$\frac{d^2 f(\lambda_j)}{d\lambda_j^2} = -\frac{T\mu\left(2PT\mu + 2P\mu^2 + T^2\right)}{\left(P\lambda_j\mu + T\lambda_j - T\mu\right)^3}.$$

As observed, the nominator is negative in the given domains and showing the denominator being negative completes the proof. Therefore, $P\lambda_j\mu + T\lambda_j - T\mu < 0$ must hold. After simple algebraic operation, we can show that $\lambda_j - \frac{T\mu}{T+P\mu} \overset{?}{<} 0$ and prove that it is true because we require $\lambda_j < \frac{T\mu}{T+P\mu}$ in the theorem. Therefore, the convexity proof is complete. □

Because of the convexity in $f(\lambda_j)$, $A+B\lambda_j$ supports $f(\lambda_j)$ at $\lambda_j = \tilde{\lambda}_j$, where $A = f(\tilde{\lambda}_j) - B\tilde{\lambda}_j$, $B = \frac{df(\tilde{\lambda}_j)}{d\tilde{\lambda}_j}$. Hence, $f(\lambda_j) \ge A + B\lambda_j - \mathbb{M}_3(1 - x_j)$ is a valid cut for any $j \in \mathcal{I}$. Solving the initial program and iteratively adding the cut constraint set (18) will guarantee a convergence to the global optimality. The large number constant, $\mathbb{M}_3$, should satisfy $\mathbb{M}_3 \ge \max_{j\in\mathcal{I}} f(\lambda_j) = \max\left(f(\sum_{d\in\mathcal{D}} 2\lambda_d^{\kappa}), f\left(\frac{ST\mu}{T+P\mu}\right)\right)$.

$$L_{j'} \ge A_{jn} + B_{jn}\lambda_{j'} - \mathbb{M}_3\left(1 - x_{j'}\right) \qquad \forall j, j' \in \mathcal{I}. \tag{18}$$

At iteration n, solving the program yields a solution vector $(\mathbf{b}, \mathbf{e}, \mathbf{g}, \mathbf{r}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\lambda}, \mathbf{L})$ and a lower bound for the objective value $\mathbf{C}_n$. An upper bound, $\widehat{\mathbf{C}}_n$, can be computed with $\mathbf{C}$ in (13) using (1) supported by the recently found $\boldsymbol{\lambda}$ vector input. Once the gap between $\mathbf{C}_n$ and $\widehat{\mathbf{C}}_n$ is satisfactorily low (i.e., approaches to zero), the algorithm terminates and the optimal system cost is $\mathbf{C}^* = \widehat{\mathbf{C}}_n$. Algorithm 1 shows the algorithmic details. The constant, Ψ, denotes the maximum allowed gap between the bounds. Although we do not explicitly show in the algorithm, the indices j and j' in (18) imply that a cut by A and B coefficients of a given j is introduced to all of the locations at each iteration. Otherwise, one may find $x_j = 1$ and $x_k = 0$ for arbitrary vertices of $j \in \mathcal{I}$ and $k \in \mathcal{I}$ in an iteration and $x_j = 0$ and $x_k = 1$ in a following iteration. Hence, the ABSM location decisions may bounce from one vertex to another leading to an increase in iteration counts toward reaching to the global optimality. As the problem (even for a small-sized one) is quite difficult and solvers require long computational times to solve, our algorithm

Table 3
Parameter values and distributions.

C_j^ϕ ($/yr)	C^δ ($/min)	C^τ ($/mile)	C^θ ($/yr)	C^β ($/yr)	P	R (min)	S	T (h)	λ_d^κ (unit/day)	μ (unit/day)
$U\,(60{,}000, 80{,}000)$[a]	0.016	0.8	2,122	122	1.0×10^{-5}	10	0.999	3	$U\,(1,130)$	480

[a]The annual ABSM cost is ∼$50,000 and the rest of the amount accounts for the space and maintenance costs.

Table 4
Cost deviation, optimality gap, and computational time summary.

Avg. cost deviation	Avg. gap	$\mathbb{B}$—Avg. gap	Avg. time (s)	$\mathbb{B}$—Avg. time (s)	Min. time (s)	$\mathbb{B}$—Min. time (s)	Max. time (s)	$\mathbb{B}$—Max. time (s)
8.08×10^{-3}	7.9×10^{-3}[a]	4.3×10^{-5}	617.96	6.80	5.21	3.87	1800	9.62

[a]The average gap is larger than the specified $\Psi = 0.001$ threshold because solution time for two instances reached 1800 s.

focuses on reducing the iteration counts rather than adding less number of cuts at each iteration.

```
Algorithm 1: An exact solution algorithm
Initialize with $n = 0$, solve the initial program, and obtain $\mathbf{C}_{n=0}$ and $\widehat{\mathbf{C}}_{n=0}$.
while $\left(\widehat{\mathbf{C}}_n - \mathbf{C}_n\right)/\widehat{\mathbf{C}}_n \geq \Psi$, do
    Set $n = n + 1$, introduce (18), solve the extended program, obtain $\mathbf{C}_n$, and compute $\widehat{\mathbf{C}}_n$.
end
Set $\mathbf{C}^* = \widehat{\mathbf{C}}_n$.
```

4. Numerical results

This section provides a brief explanation on the data used throughout the numerical studies, proposes an alternative modeling approach and compares its performance to the one earlier presented, reveals test results showing the efficiency boundaries of the exact solution method, implements the model and the solution method on a case study, and analyzes the impact of individual cost parameters (C_j^ϕ, C^δ, C^θ, C^β) and the service rate (μ) on the optimal system cost $\mathbf{C}^*$.

4.1. Data structure

The data, summarized in Table 3, are gathered from a partner enterprise, Asylon, Inc [23], and the literature [9,12]. In the table and hereafter, we use $U(\cdot)$ to denote the Uniform distribution. All time units in the table are converted to a daily planning horizon. Apart from the parameters in the table, we use Euclidean distances to estimate D_d and T_{ij} by assuming a constant 60 mph speed for drones. Cokyasar et al. [9] and Shavarani et al. [12] provided detailed justifications into these parameter values. We use these values unless otherwise stated. Finally, all computations were carried out on an Intel® Xeon® E5-1650 v3 CPU@3.5 GHz workstation with 128 GB of RAM and problem instances were solved using the Python 3.7 interface to the commercially available solver Gurobi 9.0.2 Gurobi Optimization, LLC [35].

4.2. Computational results

In this section, we compare the computational time and the solution quality with an altered version of the model in which we impose integrality conditions on the ratio of demand that can be delivered by the two mode-choices. The purpose of this experiment is to observe whether integrality conditions reduce the computational time at an acceptable solution quality loss. In the altered model, henceforth called the *benchmark model* and symbolized by $\mathbb{B}$, we consider $z_d \in \{0,1\}$, remove (4) and (15), replace g_{ijd} with y_{ijd} throughout and (3) with (19), and solve the resulting program with the same algorithm.

$$\sum_{i\in\{i|i\in\mathcal{I}\wedge T_{id}\leq R\}} y_{idd} + z_d = 1 \qquad \forall d \in \mathcal{D}, \tag{19}$$

Table 5
Computational performance of the exact solution method.

$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{D}\rvert$	Average gap	Average time (s)
40	5	1.3×10^{-4}	19.49
	10	2.3×10^{-4}	270.67
	20	8.7×10^{-2}	3600
60	5	1.6×10^{-4}	83.88
	10	1.6×10^{-2}	2202.59
	20	1.4×10^{-1}	3600

We use 100 test instances with $|\mathcal{I}| = 40$ and $|\mathcal{D}| = U(5,15)$, on the basis of the aforementioned data structure and solve them using both models. The computational time is capped by 1800 s and the maximum allowed solution gap (Ψ) is 1.0×10^{-3}. Table 4 provides a summary of the cost deviation, optimality gap, and computational time results comparing the two models. The cost deviation is the gap between $\mathbf{C}^*$ and $\mathbf{C}^*_{\mathbb{B}}$, where $\mathbf{C}^*_{\mathbb{B}}$ denotes the optimal system cost of the benchmark model. The results reveal that the benchmark model converges to the desired optimality level much quicker with a quite low cost deviation. Although a relaxation on the integrality conditions often yield a shorter time to solution, we observe an adverse relation. The reason behind is that our relaxation (in the original model) considerably changes the model and expands the solution space by introducing a larger number of feasible solutions.

Using the model presented and previously provided data structure, capping the computational time by an hour, and setting $\Psi = 5.0\times10^{-4}$, ten instances for each pair of $\mathcal{I}$ and $\mathcal{D}$ were solved to investigate the computational efficiency of the exact solution method. Table 5 illustrates the average gaps and computational times. The solution method performs quite well for small-sized instances, such as $|\mathcal{I}| = \{40, 60\}$ and $|\mathcal{D}| < 20$. In large problem sizes, the method cannot help converging to optimality in a short amount of time but produces moderate quality results, such as 8.7% gap in an hour. It should be noted that the problem is quite complex and the model is rather comprehensive that includes both facility (i.e., ABSM) location and drone routing aspects subject to strict nonlinear queueing constraints. Therefore, high computational times are a natural outcome of the complexity in the modeling.

4.3. Case study

The model and the solution method were implemented in a portion of the Chicago metropolitan area to show its applicability in real-life cases. First, the shape file – including zonal level location and population information – was extracted from a simulation tool called Polaris that is devoted to quantify the mobility and energy impacts of new technologies [36]. Then, an FC location, 32 uniformly-spaced candidate ABSM locations and 15 random (from a uniform distribution) demand vertices were selected. The daily demand rate per person is 1.051737×10^{-2}, and λ_d^κ is estimated by the household population and this daily demand rate. Other than the demand, we followed the same parameter values provided in the data structure section. The resulting

Table 6
Summary of case study solution metrics.

Number of selected ABSMs	Number of drone-delivered demand	Number of truck-delivered demand	$\sum_{j\in\mathcal{J}} L_j$	System-wide battery inventory	w
12	663.5	309.6	83.7	294.32	122.65

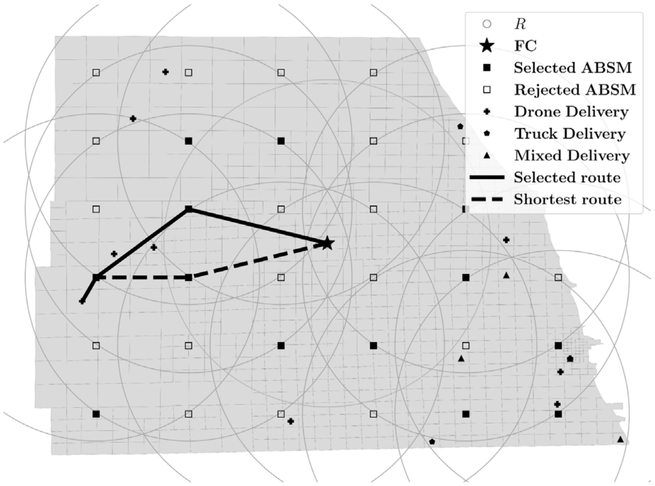


Fig. 1. Case study solution visualization.

problem was solved to 3.6×10^{-4} optimality gap in 2.7 h and the visual representation of the solution is provided in Fig. 1. The large circles represent the maximum travel range without battery swapping, i.e., R in distance units. The routes to vertices served by drones can be deduced by observing the selected ABSM locations. Although the solution yields routes to all drone-served demand vertices within $\mathbf{y}$ variables, we only show one of them to illustrate the significance of the queueing. In this figure, a selected drone delivery route and the shortest-route from the FC to the subject demand vertex are also depicted. Although a longer distance is traveled, such a routing brings in cost savings due to the decreased congestion at the ABSM location.

Key solution metrics are summarized in Table 6. To further analyze the cost savings of the proposed methodology, we investigated the optimal cost by enforcing all deliveries to be carried out by drones (i.e., $z_d = 0\ \forall d \in \mathcal{D}$) and the cost formed by using only trucks for all deliveries (i.e., $z_d = 1\ \forall d \in \mathcal{D}$). The methodology provides 12% and 19.8% cost savings compared to the drone-only and truck-only scenarios, respectively.

To encourage future research in the subject, we publicly share the data and solution files for the case study at

https://gitlab.com/tcokyasar/optimization-of-battery-swapping.

4.4. Sensitivity analyses

Since the drone technology is advancing, a decrease in the cost components and an increase in the ABSM service rate can be observed. Therefore, we investigate the impact of C_j^{ϕ}, C^{δ}, C^{θ}, C^{β}, and μ on the optimal system cost, $\mathbf{C}^*$, through sensitivity analyses. We solve ten small problem instances with $|\mathcal{J}| = 40$ and $|\mathcal{D}| = 5$ using $\Psi = 5 \times 10^{-4}$ and $P = 1.0 \times 10^{-5}$. After solving the baseline instances, the aforementioned parameters were improved up to 50% and the resulting instances were solved. The instances are specifically selected to be small to be able to solve to optimality in an acceptable time frame. The average solution time and the average optimality gap are 16.71 s and 1.49×10^{-4}, respectively. Fig. 2a illustrates the impact of percent improvement in the parameters on $\mathbf{C}^*$. In this figure, $[\mathbf{C}^*|C^{\beta}]$ and other similar notations denote the average system cost with respect to the associated parameter. Note that each marker represents the average system cost of ten instances. It is observed that C_j^{ϕ}, followed by C^{δ}, is the most critical parameter toward cost reduction. The parameters, C^{θ} and μ, impact the $\mathbf{C}^*$ in a similar rate, while C^{β} impacts the least. To further investigate the impact of battery inventory, a similar analysis was carried out by increasing P up to 3.0×10^{-5} and the results are illustrated in Fig. 2b. In these instances, further increasing P did not alter $\mathbf{C}^*$ because drone delivery became economically infeasible at $P = 2.75 \times 10^{-5}$. This graph shows that fully-charged battery availability is quite significant in the inclusion of drone delivery in the optimal system design.

5. Conclusion

In this study, we propose a modeling approach to solve the E-CDD problem, develop an exact solution method, address the capabilities of the solution method, investigate the computational efficiency, provide a case study, quantify the potential cost savings can be gained by utilizing the technology and the proposed model, and analyze the impact of parameters on the optimal system cost.

The E-CDD problem is briefly to determine: (1) the strategic locations of battery swapping machines, (2) the delivery-mode choices of a given set of demand vertices, and (3) drone routes to demand locations, while minimizing the overall system cost. An MINLP model is developed to solve the problem and an exact solution method based on cutting-planes that allows relaxing the model to an MIQCP was developed. Iteratively solving MIQCPs by introducing cutting-planes guarantees convergence to global optimality in finite number of iterations. Our computational experiments showed that the proposed solution method is the most useful in cases where $|\mathcal{J}| < 60$ and $|\mathcal{D}| < 20$. When the problem size is larger, solving each MIQCP takes very long computational times, e.g., days to weeks.

In the case study, we illustrated the applicability of our methodology and also quantified that our approach can yield 12%–20% cost savings compared to a drone-only or a truck-only delivery system. Furthermore, the sensitivity analyses investigated future scenarios considering the technology advancements and found that C_j^{ϕ} has the highest impact on the overall system cost and the delivery network design. Different from the literature, the model incorporated the cost of battery inventory and the analysis showed that battery availability at swapping locations is a key component for drone delivery to economically compete with trucks.

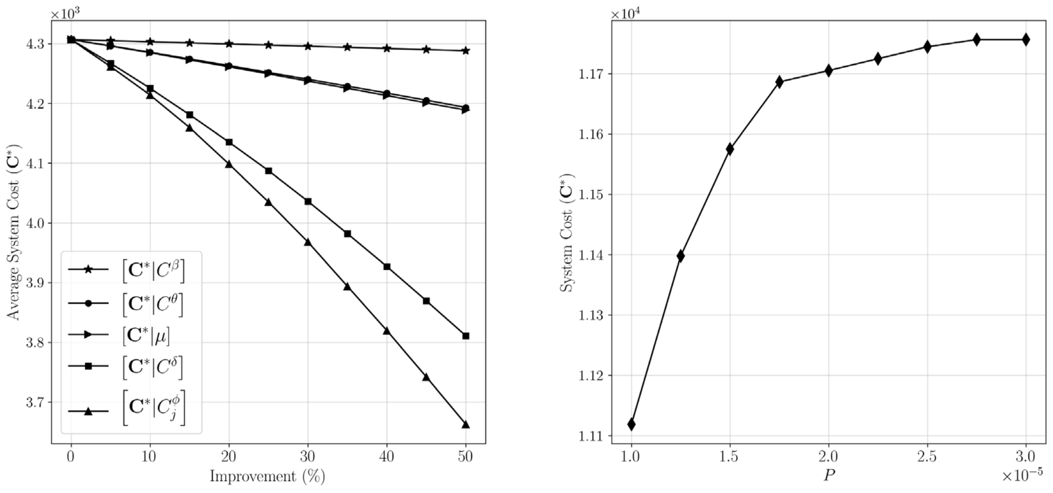
Future studies will focus on developing sound heuristic methods to solve the problem in a shorter time with an acceptable solution quality. Furthermore, an extension with multiple facilities will be developed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Vehicle Technologies Office, under the Systems and Modeling for Accelerated Research in Transportation Mobility Laboratory Consortium, an initiative of the Energy Efficient Mobility Systems Program. David Anderson, a Department of Energy, Office of Energy Efficiency and Renewable Energy manager, played an important role



(a) (b)

Fig. 2. System cost with respect to (a) C^ϕ_j, C^δ, C^θ, C^β, and μ and (b) P.

in establishing the project concept, advancing implementation, and providing ongoing guidance.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. http://energy.gov/downloads/doe-public-access-plan.

References

[1] Amazon, Inc, Amazon prime air, 2020, Available at http://www.amazon.com/primeair, accessed on Aug. 14, 2020.

[2] D. Elliot, DHL testing delivery drones, 2013, Available at https://www.cbsnews.com/news/dhl-testing-delivery-drones/, accessed on Aug. 14, 2020.

[3] A.C. Madrigal, Inside Google’s secret drone-delivery program, 2014, Available at https://www.theatlantic.com/technology/archive/2014/08/inside-googles-secret-drone-delivery-program/379306/, accessed on Aug. 14, 2020.

[4] F. Berminghman, FedEx researching drone delivery but not for widespread use, 2014, Available at https://www.ibtimes.co.uk/fedex-researching-drone-delivery-not-widespread-use-1471063, accessed on Aug. 14, 2020.

[5] R. Aitken, What do China’s delivery drones look like? – JD.com spotlight, 2017, Available at http://unmannedcargo.org/chinese-delivery-drones/, accessed on Aug. 14, 2020.

[6] FAA, Summary of Small Unmanned Aircraft Rule (Part 107), Federal Aviation Administration, Washington, DC, 2016, Available at https://www.faa.gov/uas/media/Part_107_Summary.pdf, accessed on Aug. 14, 2020.

[7] J.C. Curlander, A. Gilboa-Amir, L.M. Kisser, R.A. Koch, R.D. Welsh, Multi-level fulfillment center for unmanned aerial vehicles, 2017, US Patent 9,777,502.

[8] T. Cokyasar, W. Dong, M. Jin, Network optimization for hybrid last-mile delivery with trucks and drones, in: IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 2019.

[9] T. Cokyasar, W. Dong, M. Jin, I.O. Verbas, Designing a drone delivery network with automated battery swapping machines, Comput. Oper. Res. (2020) http://dx.doi.org/10.1016/j.cor.2020.105177, Pre-print available at https://www.sciencedirect.com/science/article/abs/pii/S030505482030294X, accessed on Dec. 24, 2020.

[10] I. Hong, M. Kuby, A.T. Murray, A range-restricted recharging station coverage model for drone delivery service planning, Transp. Res. C 90 (2018) 198–212, http://dx.doi.org/10.1016/j.trc.2018.02.017.

[11] H. Huang, A.V. Savkin, A method of optimized deployment of charging stations for drone delivery, IEEE Trans. Transp. Electr. 6 (2020) 510–518, http://dx.doi.org/10.1109/TTE.2020.2988149.

[12] S.M. Shavarani, S. Mosallaeipour, M. Golabi, G. İzbirak, A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system, Comput. Oper. Res. 108 (2019) 57–68, http://dx.doi.org/10.1016/j.cor.2019.04.001.

[13] N. Agatz, P. Bouman, M. Schmidt, Optimization approaches for the traveling salesman problem with drone, Transp. Sci. 52 (2018) 965–981, http://dx.doi.org/10.1287/trsc.2017.0791.

[14] P. Bouman, N. Agatz, M. Schmidt, Dynamic programming approaches for the traveling salesman problem with drone, Networks 72 (2018) 528–542, http://dx.doi.org/10.1002/net.21864.

[15] J.C. de Freitas, P.H.V. Penna, A variable neighborhood search for flying sidekick traveling salesman problem, Int. Trans. Oper. Res. 27 (2020) 267–290, http://dx.doi.org/10.1111/itor.12671.

[16] M. Dell’Amico, R. Montemanni, S. Novellani, Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem, Optim. Lett. (2019) http://dx.doi.org/10.1007/s11590-019-01492-z.

[17] Q.M. Ha, Y. Deville, Q.D. Pham, M.H. Hà, On the min-cost traveling salesman problem with drone, Transp. Res. C 86 (2018) 597–621, http://dx.doi.org/10.1016/j.trc.2017.11.015.

[18] C.C. Murray, A.G. Chu, The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, Transp. Res. C 54 (2015) 86–109, http://dx.doi.org/10.1016/j.trc.2015.03.005.

[19] C.C. Murray, R. Raj, The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones, Transp. Res. C 110 (2020) 368–398, http://dx.doi.org/10.1016/j.trc.2019.11.003.

[20] H. Huang, A.V. Savkin, C. Huang, Reliable path planning for drone delivery using a stochastic time-dependent public transportation network, IEEE Trans. Intell. Transp. Syst. (2020) 1–10, http://dx.doi.org/10.1109/TITS.2020.2983491, Pre-print available at https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9058989, accessed on Aug. 14, 2020.

[21] H. Huang, A.V. Savkin, C. Huang, Scheduling of a parcel delivery system consisting of an aerial drone interacting with public transportation vehicles, Sensors 20 (2020) 2045–2063, http://dx.doi.org/10.3390/s20072045.

[22] H. Huang, A.V. Savkin, C. Huang, A new parcel delivery system with drones and a public train, J. Intell. Robot. Syst. (2020) http://dx.doi.org/10.1007/s10846-020-01223-y.

[23] Asylon, Inc, Asylon, Inc, An aerial infrastructure company, 2020, Available at https://www.flyasylon.com, accessed on Aug. 14, 2020.

[24] J.E. Kelley Jr., The cutting-plane method for solving convex programs, J. Soc. Ind. Appl. Math. 8 (1960) 703–712, http://dx.doi.org/10.1137/0108053.

[25] V. Hassija, V. Saxena, V. Chamola, Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory, Comput. Commun. 149 (2020) 51–61, http://dx.doi.org/10.1016/j.comcom.2019.09.021.

[26] F. Al-Turjman, H. Zahmatkesh, I. Al-Oqily, R. Daboul, Optimized unmanned aerial vehicles deployment for static and mobile targets’ monitoring, Comput. Commun. 149 (2020) 27–35, http://dx.doi.org/10.1016/j.comcom.2019.10.001.

[27] N.A. Khan, N. Jhanjhi, S.N. Brohi, R.S.A. Usmani, A. Nayyar, Smart traffic monitoring system using unmanned aerial vehicles (UAVs), Comput. Commun. 157 (2020) 434–443, http://dx.doi.org/10.1016/j.comcom.2020.04.049.

[28] B. Rabta, C. Wankmüller, G. Reiner, A drone fleet model for last-mile distribution in disaster relief operations, Int. J. Disaster Risk Reduct. 28 (2018) 107–112, http://dx.doi.org/10.1016/j.ijdrr.2018.02.020.

[29] S. Aggarwal, N. Kumar, Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, Comput. Commun. 149 (2020) 270–299, http://dx.doi.org/10.1016/j.comcom.2019.10.014.

[30] S.S. Bacanli, D. Turgut, Energy-efficient unmanned aerial vehicle scanning approach with node clustering in opportunistic networks, Comput. Commun. 161 (2020) 76–85, http://dx.doi.org/10.1016/j.comcom.2020.07.010.

[31] P. Grippa, D.A. Behrens, F. Wall, C. Bettstetter, Drone delivery systems: Job assignment and dimensioning, Auton. Robots 43 (2019) 261–274, http://dx.doi.org/10.1007/s10514-018-9768-8.

[32] T. Kirschstein, Comparison of energy demands of drone-based and ground-based parcel delivery services, Transp. Res. D 78 (2020) 102209–102227, http://dx.doi.org/10.1016/j.trd.2019.102209.

[33] K.C. Madan, M.F. Saleh, On M/D/1 queue with general server vacations, Internat. J. Inform. Management Sci. 12 (2001) 25–38.

[34] H.-Y. Mak, Y. Rong, Z.-J.M. Shen, Infrastructure planning for electric vehicles with battery swapping, Manage. Sci. 59 (2013) 1557–1575, http://dx.doi.org/10.1287/mnsc.1120.1672.

[35] Gurobi Optimization, LLC, Gurobi Optimization, LLC, Gurobi optimizer reference manual, 2020, Available at http://www.gurobi.com, accessed on Aug. 14, 2020.

[36] J. Auld, M. Hope, H. Ley, V. Sokolov, B. Xu, K. Zhang, POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations, Transp. Res. C 64 (2016) 101–116, http://dx.doi.org/10.1016/j.trc.2015.07.017.