Computers and Operations Research 129 (2021) 105177

Contents lists available at ScienceDirect

Computers and Operations Research

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

Designing a drone delivery network with automated battery swapping machines

Taner Cokyasar [a,*], Wenquan Dong [b], Mingzhou Jin [b], İsmail Ömer Verbas [a]

[a] Energy Systems Division, Argonne National Laboratory, Lemont, IL, United States
[b] Department of Industrial and Systems Engineering, The University of Tennessee, Knoxville, TN, United States

ARTICLE INFO

Article history:
Received 23 April 2020
Revised 1 October 2020
Accepted 30 November 2020
Available online 17 December 2020

Keywords:
Drone delivery
Network optimization
Mixed-integer nonlinear programming
Queueing theory

ABSTRACT

Drones are projected to alter last-mile delivery, but their short travel range is a concern. This study proposes a drone delivery network design using automated battery swapping machines (ABSMs) to extend ranges. The design minimizes the long-term delivery costs, including ABSM investment, drone ownership, and cost of the delivery time, and locates ABSMs to serve a set of customers. We build a mixed-integer nonlinear program that captures the nonlinear waiting time of drones at ABSMs. To solve the problem, we create an exact solution algorithm that finds the globally optimal solution using a derivative-supported cutting-plane method. To validate the applicability of our program, we conduct a case study on the Chicago Metropolitan area using cost data from leading ABSM manufacturer and geographical data from the planning and operations language for agent-based regional integrated simulation (more commonly known as POLARIS). A sensitivity analysis identifies that ABSM service times and costs are the key parameters impacting the long-term adoption of drone delivery.

1. Introduction

Unmanned aerial vehicles (UAVs), more commonly known as *drones*, date back to 1840s in the form of non-controllable hot air balloons (Buckley, 2006). According to McKenna (2016), the British Royal Air Force carried out one of the earliest applications of controllable UAVs in 1935. The term "drone," inspired by male bees, is believed to have first been used by the U.S. Navy in around 1935 (Clarke, 2014). Thereafter, UAVs have served the military for a variety of missions. Apart from being used for recreational purposes, most recently, drones are planned to be adopted as an alternative last-mile delivery mode. Amazon, Inc did the first prime air delivery through their Amazon Prime Air program in 2016 (Amazon, Inc, 2020). Meanwhile, the U.S. Government, through the Federal Aviation Administration (FAA), released regulations under FAA Part 107 to limit drone operations and has granted waivers to individuals and enterprises, allowing them to carry out research in the area (FAA, 2016). Further, FAA-certified providers have recently built a low altitude authorization and notification capability (i.e., a flight traffic control mechanism) to allow safe drone operations.

Amazon filed a patent showing a beehive-shaped multi-level distribution center (DC) design they plan to build to ease and automate landing and take off (Curlander et al., 2017). Under the current state of the battery technology, Amazon and some other early adopters could build these high-cost facilities at a distance of every 10 miles in densely populated areas to carry out parcel delivery services. Also, automated battery swapping machines (ABSMs) were proposed as a range extension approach to minimize the number of high-cost DCs (Cokyasar et al., 2019; Hong et al., 2018). Asylon, Inc (2020) produces ABSMs for use in autonomous security patrols, inventory tracking, thermal inspection, etc. The machine automatically replaces batteries once a drone lands on the platform and stacks the replaced battery to an integrated battery charging dock. Thus, the technology is commercially available, although alteration and improvement might be needed to construct an Amazon-scale drone delivery network. An obvious benefit of the machine is allowing drones fly longer distances. This delivery network is similar to a telecommunication network, where towers are placed at certain distances to maximize service coverage while accounting for the service load. Similar considerations apply to this context as well due to constant battery swap operational time and battery limited drone flight ranges.

Since drones are projected to be an alternative delivery mode, one of the critical problems is to design a cost-efficient network infrastructure. We call it the drone delivery network problem (DDNP). This problem would determine delivery *modes* and drone

* Corresponding author.
E-mail addresses: tcokyasar@anl.gov (T. Cokyasar), dwenquan@utk.edu (W. Dong), jin@utk.edu (M. Jin), omer@anl.gov (İ.Ö. Verbas).

https://doi.org/10.1016/j.cor.2020.105177
0305-0548/

routes of a customer set and the number of drones to employ while minimizing the long-term operational cost of the network. Based on Amazon's beehive-shaped DC design patent, alternative delivery modes could be *solely* trucks or drones. We explicitly use the term "solely" because a few recentstudies proposed the use of drones coupled with truck deliveries (Murray and Chu, 2015; Ha et al., 2018), and others studied drones coupled with public transportation vehicles and trains (Huang et al., 2020a,b). How delivery drones will be deployed in the future is uncertain, and researchers are proposing different methods to build a theoretical infrastructure.

Our study considers sole truck or sole drone deliveries for each customer as a delivery mode and focuses on a single DC as the current state of the technology and business models do not support deployment of this delivery network at large-scale. A drone delivery route to a customer connects the DC, ABSM locations, and the customer. To form possible drone delivery routes, DDNP requires deciding where to house ABSMs out of a given set of candidate locations.

In this study, we build a mixed-integer nonlinear programming (MINLP) model to solve the DDNP and partially fill the gap in the literature by introducing an alternative delivery layout and developing an exact solution algorithm. For package delivery by trucks in a city, the schedule and routes are rather consistent over days and is independent from whether there is one more or one fewer customer. We expect that drone delivery–initially–will only be adopted for expedited orders and comprises a very small percent of total deliveries due to its high investment cost and limited capacity. Many models and algorithms have been developed for optimizing delivery routes for trucks and employed by companies, such as UPS (Holland et al., 2017). Therefore, in DDNP, we assume that truck routing problem has been solved, and the cost for delivering a package to a specific customer by a truck is known beforehand, which is mainly related to lead time caused by the fixed route and schedule rather than the distance between the customer and the depot. The operational cost and the mode choice decisions are affected by operational time (including customer wait times), considering that a faster delivery of goods can improve customer loyalty and drive high profitability in the long-term. Hence, we employ the queueing theory in our program and the nonlinearity is a natural outcome of this consideration.

Our contribution to the literature is listed as follows. We introduce an MINLP to model the DDNP to minimize the long-term operational cost and develop an exact solution approach based on the cutting-plane method (Kelley, 1960). We provide computational experiments addressing the capabilities of our solution method, conduct a case study to show the applicability of our model, and provide a sensitivity analysis to assess the impact of key parameters on the optimal network configuration.

The rest of the paper is organized as follows. In Section 2, we provide a literature review to identify the similarities and differences of our problem, modeling, and solution approaches with other drone delivery studies. We describe the problem and present the MINLP in Section 3. In Section 4, we provide linearization procedures and an exact solution algorithm to solve the DDNP. Section 5 summarizes the data layout and illustrates computational results, a case study, and a sensitivity analysis. Finally, we mention model limitations and conclude the study in Section 6.

2. Literature review

In the current drone delivery literature, many studies focus on variants of "drone-assisted parcel delivery" (DAPD) problems (Murray and Chu, 2015) as the state of the technology and the U. S. government regulations constrain the operation of full-automated drone deliveries (Murray and Chu, 2015; Agatz et al., 2018; Bouman et al., 2018; Ha et al., 2018; Dell'Amico et al., 2019; Kitjacharoenchai et al., 2019; de and Penna, 2020; Murray and Raj, 2020). In DAPD, trucks are driven on an optimized route and drones simultaneously fly the so-called *last-mile* (i.e., a predetermined short distance) to the consumer's door to deliver packages. The DAPD problem, first described in Murray and Chu (2015), requires optimizing the routes of the delivery vehicles to minimize the overall operational cost. As Murray and Raj (2020) mentioned, the DAPD serves as a viable and economical first step transition to lower delivery costs during the drone technology advancement phase. Yet, the DAPD does not promise full-automation, and alternative approaches are required to maximize the benefit gained from the emerging technology. As seen in Curlander et al. (2017), full-automation would allow sole drone deliveries. In our study, rather than coupling trucks and drones, we use them as sole competitive delivery modes for each customer and construct a long-term drone delivery infrastructure with ABSMs. Since the idea is not as close as DAPD to realization, there are only a few studies to date (Hong et al., 2018; Cokyasar et al., 2019; Shavarani et al., 2019).

To the best of our knowledge, the idea of using ABSMs to create a drone delivery network was first published by Hong et al. (2018). Soon after, Cokyasar et al. (2019) proposed an alternative approach to site ABSMs. In Hong et al. (2018), the objective is to maximize the area covered by drones using a single DC and multiple ABSMs. Their study focuses solely on drones and does not aim to determine the optimal delivery mode for each customer location. More importantly, their study does not consider a possible congestion at ABSMs caused by high number of drones requesting battery swap service concurrently. On the other hand, Cokyasar et al. (2019) identifies this as a bottleneck in the delivery system and minimizes the overall delivery cost with a focus on the delivery time impacted by concurrent services considering a single DC, multiple ABSMs, and a *single-hop*. Hopping can be defined as a capability that enables drones to reach farther distances by replacing their batteries with fully-charged ones at ABSM locations or recharging batteries at designated locations. Hence, single-hop refers to a route starting at a DC, stopping by an ABSM, and delivering a parcel to a customer location, while multiple-hop allows stopping by multiple ABSMs along the one-way delivery route. This study is built upon Cokyasar et al. (2019) by reconstructing an optimization model that covers the multiple-hop aspect of the problem, bringing in the drone equipment ownership costs into the decision-making process, and introducing an improved exact solution method benefiting from convexity properties.

Other studies related to drones focused on border surveillance (Kim and Lim, 2018), flight scheduling under battery duration uncertainty (Kim et al., 2018), battery assignment and scheduling (Park et al., 2017), and medication deliveries (Scott and Scott, 2019).

Following the stochastic queue median (SQM) literature, Shavarani et al. (2019) presented an $M/G/k$ queue embedded MINLP model using fuzzy variables to represent uncertainties in demand and drone travel time. Although their problem definition is quite close to ours, they present a completely different modeling approach and solve the problem using heuristic algorithms that do not always promise a globally optimal solution. In their model setting, they consider drones as *mobile* servers and assume the service time (i.e., the travel time from the DC to a customer including battery swap stop times on the route) follows a general distribution. Hence, each order would be exposed to a virtual queue on the e-commerce retailer's system. In our setting, we consider *immobile* (stationary) ABSMs as servers [as in] (Wang et al., 2002) with

deterministic battery swap times and assume the travel time is fixed. Our model also considers trucks as an alternative delivery mode while their model focus solely on drones. To be fair, we acknowledge that their model is more comprehensive in some aspects (e.g., it considers battery costs and multiple variable DC locations). On the other hand, we construct a tractable MINLP that can be solved to optimality.

We shall also provide some details on the SQM literature as our modeling approach somewhat resembles the models in this literature. SQM problem considers optimizing the location(s) of server(s) so that the expected response times to service requests at vertices are minimized. Most papers in this literature consider mobile servers and exponentially or generally distributed service times. While our model in part simplifies the problem by using immobile servers, our challenge comes in the form of routing. In other words, to allow drone usage for customers in farther distances (from the DC), our model should ensure siting (opening) ABSMs along the path. Such a routing requires balancing the queue lengths (i.e., the number of drones waiting in the line of an ABSM to receive battery swap service) not only on a single vertex but along the paths. Otherwise, battery swap service accumulation at ABSMs can occur due to the stochastic nature of the e-commerce demand. Briefly, in SQM literature, Larson (1974) presented the first queue involving location problem. Studies by Berman et al. (1985, 1987), Chiu and Larson (1985), Batta (1989) and Batta and Berman (1989) were the early works about different variations of the problem. Berman and Drezner (2007) and Hamaguchi et al. (2010) presented more recent studies. We omit providing further details on SQM literature because our model does not one-to-one match to any of those listed above, but we refer the reader to a review article (Snyder, 2006).

3. Mathematical model

We can describe the DDNP as follows. To begin with, let $(\mathcal{V}, \mathcal{E})$ represent a delivery network with vertices and edges, respectively, and Table 1 defines all the sets and parameters. Given a single DC location that operates the delivery service of a customer set $\mathcal{K}$ and a given candidate ABSM location set $\mathcal{I}$, find the optimal 1) network configuration by determining the delivery mode of demand $k \in \mathcal{K}$, 2) locations in $\mathcal{I}$ to house ABSMs, 3) number of drones needed for the system to seamlessly operate in long-term, and 4) utilization rate at $j \in \mathcal{I}$ to avoid a bottleneck in the network, while minimizing the long-term average system costs. We consider two direct delivery modes: drones and trucks. As cost components, we utilize ABSM, drone ownership, drone and truck delivery, and drone delivery lead time costs.

We assume that the demand for each customer location $k \in \mathcal{K}$ follows a Poisson process with a rate λ_k, only one unit is demanded at a time by $k \in \mathcal{K}$, and each drone can deliver one unit of demand during a single delivery. Delivering one unit in a single delivery is a common industry practice in the development phase of the drone delivery systems.We consider identical ABSMs and drones with fixed flight speed. The battery swap operation time is reportedly a constant at $1/\mu$ (Asylon, Inc, 2020). To assess the validity of our assumptions, we interviewed with Asylon, an ABSM manufacturer/operator, and received cost and other drone related data (e.g., maximum drone travel range under the speed limitations of FAA regulations and ABSM and drone ownership costs) as well as validating the assumption. Since the battery swap takes a certain amount of time and multiple (or many) drones can arrive at an ABSM location to receive swapping service in a short period due to the stochastic nature of the demand, we consider a physical queue to be (possibly) formed at each $j \in \mathcal{I}$. We assume each $j \in \mathcal{I}$ can house up to one ABSM and has adequate space to keep a large number of drones in a waiting queue. We note that our "one ABSM per $j \in \mathcal{I}$" assumption can be easily removed by considering two or more candidate sites at the same location. ABSMs are considered to possess adequate number of fully-charged batteries at any time when a drone requests a service and service downtime caused by insufficient batteries is not permitted. Further, we assume the demand for $k \in \mathcal{K}$ is met through a single delivery mode and partitioning the demand is not allowed throughout the planning horizon. We do not consider the optimal truck routing but assume this route is known and the cost of serving $k \in \mathcal{K}$ is an input. Lastly, the drone travel range is assumed to be a constant independent from the loaded weight and the weather conditions.

Under the optimal network configuration, the drone delivery system operates as follows. Once customer $k \in \mathcal{K}$ that is decided to be served by drones demands a unit of any type of product, a drone with a fully-charged battery is loaded with the unit at the DC. Then, the drone travels from the DC to $k \in \mathcal{K}$ by hopping from one ABSM location to another where its battery is replaced with a fully-charged one. After delivering the product, the drone travels from $k \in \mathcal{K}$ to the DC in a similar fashion and is allowed to take a different route on the way back. Arriving at the DC, it is equipped with a fully-charged battery to be ready for the next delivery task. In cases where a customer will be served by trucks, the delivery operation follows a traditional routine with a fixed cost for each parcel.

Fig. 1 shows how drone delivery system functions. In this figure, three customers (denoted by 1, 2, and 3 and three ABSM locations (denoted by A1, A2, and A3) are considered. Different colors are used to differentiate the routes (dashed lines) from the DC to customers, and solid circles represent the maximum travel range, half

Table 1
Sets and parameters used in the model.

	Definition
Set	
$\mathcal{V}$	set of vertices in the network, $i \in \mathcal{V} = \mathcal{I} \cup \mathcal{K} \cup \{0\}$, where $\{0\}$ denotes the DC.
$\mathcal{K}$	subset of vertices for customer locations, $k \in \mathcal{K}, \mathcal{K} \subset \mathcal{V}$.
$\mathcal{I}$	subset of vertices for battery swapping machine candidate locations, $i \in \mathcal{I}, \mathcal{I} \subset \mathcal{V}$.
$\mathcal{E}$	set of edges in the network, $(i,j) \in \mathcal{E}, \forall i,j \in \mathcal{V}, i \neq j, T^{\delta}_{ij} \leqslant R$. (Note that we also exclude (i,j) where i and j are both in $\mathcal{K}$ because such edges will never be traveled.)
$\mathcal{I}_j\{\cdot\}$	subset of neighboring vertices of a given vertex $j \in \mathcal{I}$ that can be traveled to/from without battery swapping, $\mathcal{I}_j\{\cdot\} = \left\{i \middle\| T^{\delta}_{ij} \leqslant R \forall i \in \{\cdot\}, i \neq j\right\} \forall j \in \mathcal{I}$. (Note that $\{\cdot\}$ denotes the input set for index i. We use $i \in \mathcal{V}$ or $i \in \mathcal{I} \cup \mathcal{K}$ or $i \in \mathcal{V} \setminus \mathcal{K}$ appropriately.)
$\mathcal{I}'_k$	subset of neighboring vertices of a given vertex $k \in \mathcal{K}$ that can be traveled to/from without battery swapping, $\mathcal{I}'_k = \left\{i \middle\| T^{\delta}_{ik} \leqslant R \forall i \in \mathcal{I}\right\} \forall k \in \mathcal{K}$.
Parameter	
C^{ϕ}_j	fixed cost of locating a battery swapping machine at location $j \in \mathcal{I}$ per time unit, $C^{\phi}_j \in \mathbb{R}^+$.
C^{δ}	unit drone transportation cost per time unit, $C^{\delta} \in \mathbb{R}^+$.
C^{τ}_k	truck transportation cost per demand unit to $k \in \mathcal{K}, C^{\tau}_k \in \mathbb{R}^+$
C^{θ}	drone ownership cost per time unit, $C^{\theta} \in \mathbb{R}^+$.
T^{δ}_{ij}	travel time on any origin–destination pair (i,j) in the network for drones, where$i,j \in \mathcal{V}, T^{\delta}_{ij} \in \mathbb{R}^+$. (Note that any (i,j) pair is not considered as a feasible edge due to limited travel time, R, and T^{δ}_{ij} is symmetric (i.e., $T^{\delta}_{ij} = T^{\delta}_{ji}$).
λ_k	demand rate at customer vertex $k \in \mathcal{K}$ during the planning horizon, $\lambda_k \in \mathbb{R}^+$.
μ	service rate of battery swapping machine during the planning horizon, $\mu \in \mathbb{R}^+$.
R	maximum travel time without battery swapping, $R \in \mathbb{R}^+$.

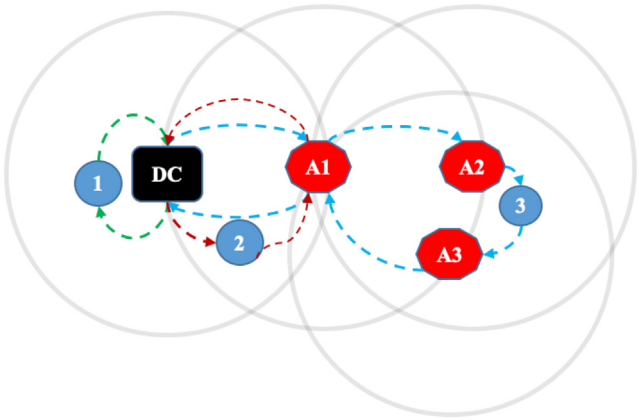


Fig. 1. An example delivery network design for drones.

of which corresponds to the service range. Customer 1 is within the service range of the DC; hence, drones can serve this customer without a battery swap. (Green arrows represent a hypothetical route to this customer.) Customer 2 is not within the service range of the DC because a drone delivering the parcel will not have adequate battery capacity to return to the DC. Therefore, the route is as follows: DC → 2 → A1 → DC. Customer 3 is served with multiple battery swaps at A1, A2, and A3. In the forward direction (from DC to customer 3), A2 is used, but in the return A3 is preferred because A2 is busy with other drones that are not displayed in the figure.

At each ABSM location, batteries are swapped following a first-come, first-served (FCFS) discipline. From a queueing theory perspective, customer demand is equivalent to a drone demanding a battery swap and ABSMs are the servers responsible for the swap operation with a constant service time denoted by $1/\mu$. Eq. 1, defining the average number of drones in the queue of ABSM at $j \in \mathcal{I}$, is a performance measure of the well-known $M/D/1$ queueing theory working with an FCFS discipline. The formulation can be found in many books devoted to the queueing theory, such as Kleinrock (1975, Chapter 5, pp. 188) and Stewart (2009, Chapter 14, pp. 517). As the multi-class Poisson demand arrivals can be aggregated into a single λ_j (the aggregate demand rate at ABSM location $j \in \mathcal{I}$) and the ABSM operation takes a deterministic $1/\mu$ time per swap, this queueing discipline is suitable for our modeling purposes. Note that $\lambda_j = \sum_{\substack{i \in \mathcal{I}_j\{\mathcal{V}\}, \\ k \in \mathcal{K}}} \lambda_k \left(y^+_{ijk} + y^-_{jik}\right) \forall j \in \mathcal{I}$ and is not explicitly constrained in our model as we incorporate it within $\rho_j = \lambda_j/\mu$.

$$L^q_j\left(\rho_j\right) = \frac{\rho_j^2}{2\left(1-\rho_j\right)} \qquad \forall j \in \mathcal{I} \tag{1}$$

The variable, ρ_j, accounts for the utilization rate at $j \in \mathcal{I}$ and y^+_{ijk} and $y^-_{jik}$are the binary routing variables to indicate whether an edge $(i,j) \in \mathcal{E}$ is traversed during the delivery service of $k \in \mathcal{K}$ along the forward and backward paths, respectively. As we allow taking different routes while going to and returning from $k \in \mathcal{K}$, we introduce two separate sets of variables, namely $\mathbf{y}^+$ and $\mathbf{y}^-$. Here, the positive sign indicates forward (from DC to customer) flights, and the negative sign refers to backward (from customer to DC) flights. While (i,j) pair refers to a flight from i to j in $\mathbf{y}^+$ variables, (j,i) pair indicates a flight from i to j in $\mathbf{y}^-$ variables. All the definitions of the decision variables can be found in Table 2.

Based on the definitions and assumptions, we develop the following MINLP to solve the DDNP.

$$\text{Minimize} \quad \mathbf{C} = \sum_{j \in \mathcal{I}} C^\phi_j x_j + \sum_{k \in \mathcal{K}} \lambda_k C^\tau_k z_k + \sum_{\substack{(i,j) \in \mathcal{E}, \\ k \in \mathcal{K}}} \lambda_k C^\delta T^\delta_{ij} \left(y^+_{ijk} + y^-_{ijk}\right) + C^\theta d + \sum_{j \in \mathcal{I}} C^\delta \left(\rho_j + L^q_j\right). \tag{2}$$

$$\sum_{i \in \mathcal{I}'_k} y^+_{ikk} = \sum_{i \in \mathcal{I}'_k} y^-_{ikk} \qquad \forall k \in \mathcal{K}. \tag{3}$$

$$\sum_{i \in \mathcal{I}'_k} y^+_{ikk} + z_k = 1 \qquad \forall k \in \mathcal{K}. \tag{4}$$

$$\sum_{i \in \mathcal{I}'_k} T^\delta_{ik} \left(y^+_{ikk} + y^-_{ikk}\right) \leqslant R \qquad \forall k \in \mathcal{K}. \tag{5}$$

$$\sum_{i \in \mathcal{I}_j\{\mathcal{V} \setminus \mathcal{K}\}} y^+_{ijk} = \sum_{i \in \mathcal{I}_j\{\mathcal{I} \cup \mathcal{K}\}} y^+_{jik}, \sum_{i \in \mathcal{I}_j\{\mathcal{V} \setminus \mathcal{K}\}} y^-_{ijk} = \sum_{i \in \mathcal{I}_j\{\mathcal{I} \cup \mathcal{K}\}} y^-_{jik} \qquad \forall j \in \mathcal{I}, k \in \mathcal{K}. \tag{6}$$

$$\sum_{\substack{i \in \mathcal{I}_j\{\mathcal{V} \setminus \mathcal{K}\}, \\ k \in \mathcal{K}}} \left(y^+_{ijk} + y^-_{ijk}\right) \leqslant \mathbb{M}_1 x_j \qquad \forall j \in \mathcal{I}. \tag{7}$$

$$d \geqslant \sum_{\substack{(i,j) \in \mathcal{E}, \\ k \in \mathcal{K}}} \lambda_k T^\delta_{ij} \left(y^+_{ijk} + y^-_{jik}\right) + \sum_{j \in \mathcal{I}} \left(\rho_j + L^q_j\right). \tag{8}$$

$$\rho_j = \mu^{-1} \sum_{\substack{i \in \mathcal{I}_j\{\mathcal{V}\}, \\ k \in \mathcal{K}}} \lambda_k \left(y^+_{ijk} + y^-_{jik}\right) \qquad \forall j \in \mathcal{I} \tag{9}$$

$$L^q_j = \frac{\rho_j^2}{2\left(1-\rho_j\right)} \qquad \forall j \in \mathcal{I} \tag{10}$$

$$x_j, y^+_{ijk}, y^-_{ijk}, z_k \in \{0,1\}, d, \rho_j, L^q_j \in \mathbb{R}^+, \ \rho_j < 1. \tag{11}$$

The objective function Eq. (2) minimizes the total cost of the system denoted by $\mathbf{C}$ for a given planning horizon. Each summation in Eq. (2) respectively considers the costs during the planning horizon for: 1) ABSMs, 2) truck delivery, 3) drone delivery, 4) drone ownership, and 5) lead time due to battery swapping operation (including the waiting time in a probable queue). Constraint set Eq. (3) ensures drones delivering the parcels to location $k \in \mathcal{K}$ leaves the location after the package is dropped off. Constraint set Eq. (4) enforces each customer location $k \in \mathcal{K}$ to be served by either a drone or a truck. Constraint set Eq. (5) guarantees the total travel time from the immediate ABSM locations (before delivery and after delivery) of $k \in \mathcal{K}$ is less than or equal to the maximum allowed travel time. Constraint set Eq. (6) balances the flow on a given vertex $j \in \mathcal{I}$, i.e., a drone flying from the origin i to deliver the demand for customer location k and stopping by a destination vertex j leaves the vertex after receiving a battery swap service. Constraint set Eq. (7) satisfies that a battery swap service at location $j \in \mathcal{I}$ can be performed only if the location is selected to house an ABSM. The constant $\mathbb{M}_1$ in this constraint satisfies $\mathbb{M}_1 \geqslant 2\max_{j \in \mathcal{I}}\left(|\mathcal{I}_j\{\mathcal{V} \setminus \mathcal{K}\}|\right)$. Constraint Eq. (8) approximates the average number of drones needed. Respectively, constraint sets Eqs. (9) and (10) define the utilization rate and the average number of drones in the battery swapping queue of candidate ABSM location $j \in \mathcal{I}$ in terms of the routing variables. In Eq. (11), we list non-negativity conditions and domains of variables.

In Eq. (5), we ensure that a drone flying from $i \in \mathcal{I}'_k$ to $k \in \mathcal{K}$ would have adequate battery capacity to make a return to $i' \in \mathcal{I}'_k$, where i' can be same as i or a different ABSM location. In other words, a drone can traverse different *last* edges while making the delivery to a customer and returning back to the DC. The constraint set Eq. (5) sums last edge variables that can be chosen for k in product with their corresponding travel time for both directions and guarantees that the total travel time on any selected edges

Table 2
Variables used in the model.

Variable	*Definition*
x_j	$\{1,$ if a battery swapping machine is located at vertex $j \in \mathcal{I}, 0,$ otherwise..
y^+_{ijk}	$\{1,$ if a drone flies on edge $(i,j) \in \mathcal{E}$ to satisfy demand at customer vertex $k \in \mathcal{K}$ (note that (i and j correspond to the origin and the destination, respectively, the first index always indicates the origin), 0, otherwise..
y^-_{jik}	$\{1,$ if a drone flies on edge $(j,i) \in \mathcal{E}$ to return the DC after meeting demand at vertex $k \in \mathcal{K}$ (note that i and j correspond to the origin and destination, respectively, and the second index always indicates the origin), 0, otherwise..
z_k	$\begin{cases} 1, & \text{if customer } k \in \mathcal{K} \text{ is served by trucks,} \\ 0, & \text{otherwise.} \end{cases}$
ρ_j	utilization rate at candidate location $j \in \mathcal{I}$, $\rho_j < 1$, $\rho_j \in \mathbb{R}^+$.
L^q_j	average number of drones at candidate location $j \in \mathcal{I}$ waiting battery swap service in the queue (excluding the one receiving service), $L^q_j \in \mathbb{R}^+$.
d	average number of drones used, $d \in \mathbb{R}^+$.

do not exceed the maximum travel time without battery swapping. We also note that the definition of $\mathcal{I}'_k$ does not satisfy this condition as it only considers one-way battery capacity adequacy, while determining neighboring vertices of a given customer vertex.

In Eq. (8), the summation $\sum_{(i,j)\in\mathcal{E},k\in\mathcal{K}} \lambda_k T^\delta_{ij}\left(y^+_{ijk} + y^-_{jik}\right)$ accounts for the average number of drones (i.e., demand units) flying in the network at any moment. The rest of the right-hand-side of the constraint, $\sum_{j\in\mathcal{I}}\left(\rho_j + L^q_j\right)$, corresponds to the average number of drones at all ABSMs. The right-hand-side, in total, approximates (due to the stochastic feature of the demand) the average number of drones used. A similar formulation can be found in a relevant study (Shavarani et al., 2019) and in a completely different study focusing on electric vehicles (Mak et al., 2013). Note that the average number of drones (d) may not be adequate to eliminate a ready-to-fly drone unavailability issue. In the model, we assumed that a drone is always available at the DC. An $M/M/s/GD/\infty/\infty$ queueing model, where s denotes the total number of drones in the system, can be used to approximate the number of drones needed to satisfy a desired level of drone availability, once the solution from our model is obtained. The model details can be found in [pp. 1087–1095] Winston (2004). Briefly, let $\lambda^\delta = \sum_{k\in\mathcal{K}} \lambda_k(1 - z_k)$ overall demand rate at the DC for all customers. Then, the average service time of drones can be defined as $1/\mu^\delta = d/\lambda^\delta$. The probability of confronting a drone unavailability when a customer order arrives is roughly

$$\frac{(s\rho^\delta)^s \pi^\delta_0}{s!(1-\rho^\delta)},$$

where $\rho^\delta = \lambda^\delta/(\mu^\delta s)$ and $\pi^\delta_0 = \frac{1}{\sum_{i=0}^{s-1}\frac{(s\rho^\delta)^i}{i!} + \frac{(s\rho^\delta)^s}{s!(1-\rho^\delta)}}$. With these equations, we can control the probability that a drone is available when a customer demand arrives by deciding the number of the total drones in the system.

In Eq. (9), the utilization rate at each ABSM is calculated by summing independent identically-distributed random parameters (λ_k) in product with their associated $\mathbf{y}^+$ and $\mathbf{y}^-$ variables. This operation is a common application adopted in a number of studies (Hamaguchi et al., 2010; Shavarani et al., 2019; Xie et al., 2018).

Clearly, Eq. (10) poses a challenge to solve the problem caused by the nonlinear terms. Although commercially available solvers such as Gurobi and IBM CPLEX can handle the nonlinear expression in Eq. (10), their computational performance is not satisfying. In the numerical results, solution time comparisons for a set of instances will be provided. In Section 4, we will address the nonlinearity and provide solution approaches to linearize and make the program bettersuitable (i.e., linear) for the available solvers.

4. An exact solution method

In this section, we explore model properties and introduce an exact solution method. First, we define set $\mathcal{T}^*$ to indicate customers that are optimally served by trucks. With this set definition, we eliminate customers from the optimization program in cases, where some customers may not be served by drones due to limited battery capacity or a drone delivery is not economically viable.

Definition 1. Customer $k \in \mathcal{T}^*$, if $T^\delta_{ik} + T^\delta_{ki'} > R \quad \forall i, i' \in \mathcal{I}'_k$.

ABSM costs (C^ϕ_j) make the delivery mode decision complicated at a customer level because choosing drone delivery for k requires opening ABSM(s) along the *path* $(0,k)$, and the same ABSM(s) can be utilized for another customer if the network topology and ABSM capacity permit. Queuing costs ($C^\delta\left(\rho_j + L^q_j\right)$) also pose a similar challenge as values for these metrics are formed based on interdependent delivery mode decisions. One way to compare the two delivery mode costs at the customer level is finding the least possible drone delivery cost. We consider the shortest-path $(0,k)$ and calculate the drone delivery cost at a customer level. Comparing this cost to the truck delivery cost reveals whether the drone delivery is an economic competitor to the truck delivery or not. Since we use the same tuple notation for both edges and paths, we will refer to (i,j) as path (i,j) where necessary. To calculate the utilization rate of each ABSM at the customer level, we define $\rho^k_j = 2\lambda_k/\mu$.

Lemma 1. *Let $\mathcal{P}_k$ contain edges $(i,j) \in \mathcal{E}$ for the shortest drone delivery path $(0,k)$. Neglecting $C^\phi_j, k \in \mathcal{T}^*$, if*

$$\left(C^\delta + C^\theta\right) \sum_{(i,j)\in\mathcal{P}_k} \left[2\lambda_k T^\delta_{ij} + \rho^k_j + L^q_j\left(\rho^k_j\right)\right] > \lambda_k C^\tau_k \text{and} \rho^k_j < 1 \forall (i,j) \in \mathcal{P}_k. \tag{12}$$

Proof. Since T^δ_{ij} is symmetric, the shortest-path edges in $\mathcal{P}_k$ for path $(0,k)$ is also the shortest-path for path $(k,0)$. Based on this fact, we write the objective function components that calculate drone and truck delivery costs of each k as follows. 1) Drone delivery cost: $C^\delta \sum_{(i,j)\in\mathcal{P}_k} 2\lambda_k T^\delta_{ij}$, 2) Drone ownership cost: $C^\theta \sum_{(i,j)\in\mathcal{P}_k} \left[2\lambda_k T^\delta_{ij} + \rho^k_j + L^q_j\left(\rho^k_j\right)\right]$, 3) Lead time cost: $C^\delta \sum_{(i,j)\in\mathcal{P}_k} \left[\rho^k_j + L^q_j\left(\rho^k_j\right)\right]$, 4) Truck delivery cost: $\lambda_k C^\tau_k$. The constant '2' in 1 and 2 indicates that drones follow the same path in both directions. Summation of 1, 2, and 3, given in Lemma 1, accounts for minimum drone delivery cost for k (excluding ABSM costs). If this minimum cost is higher than the truck delivery cost and the shortest-path drone delivery for k does not violate the utilization rate rule (i.e., $\rho < 1$) at any ABSM, this customer should be served by trucks because serving more customers through the edges in this shortest-path will not decrease individual drone delivery cost for the subject k. □

Although Lemma 1 sets a weak condition for a possible pre-assignment of $k \in \mathcal{K}$, the condition may not be helpful in situations where drone and truck delivery costs are competitive, and the ABSM cost predominantly determines the drone delivery costs.

At this point, we focus on the nonlinear term, L_j^q. The nonlinearity lies in Eq. (10) due to ρ_j. Replacing ρ_j with Eq. (9), where it is defined in terms of the routing variables, could visualize the challenge for this linearization. The following theorem sets a corner stone to be able to benefit from "the fundamental convergence theorem" of Kelley (1960) in deriving an exact solution method using a cutting-plane method.

Theorem 1. *The function* $L_j^q\left(\rho_j\right)$ *is convex in* ρ_j *in the domain of* $\rho_j \in [0,1)$.

Proof. To prove Theorem 1, we take the second order derivative of $L_j^q\left(\rho_j\right)$ with respect to ρ_j. The convexity can be proven with $\frac{d^2 L_j^q(\rho_j)}{d\rho_j^2} \geqslant 0$. As $\frac{d^2 L_j^q(\rho_j)}{d\rho_j^2} = \frac{1}{(1-\rho_j)^3} \geqslant 0$ in the domain $\rho_j \in [0,1)$, our proof is complete. □

Due to the convexity in Theorem 1, $a + b\rho_j$ supports $L_j^q\left(\rho_j\right)$ at $\rho_j = \tilde{\rho}_j$, where $a = L_j^q(\tilde{\rho}_j) - b\tilde{\rho}_j, b = \frac{dL_j^q(\tilde{\rho}_j)}{d\tilde{\rho}_j}$. Hence, $L_j^q\left(\rho_j\right) \geqslant a + b\rho_j - \mathbb{M}_2(1 - x_j)$ is a valid cut for any $j \in \mathcal{I}$. Solving the program Eqs. (2)–(9), (which will be henceforth called *the program*) by iteratively adding the cut Eq. (13) will guarantee a convergence to the global optimality. Note that the big number constant, $\mathbb{M}_2$, satisfies $\mathbb{M}_2 \geqslant L_j^q(\bar{\rho})$ for any arbitrary $j \in \mathcal{I}$, where $\bar{\rho} \to 1$.

$$L_{j'}^q \geqslant a_{jr} + b_{jr}\rho_{j'} - \mathbb{M}_2(1 - x_{j'}) \qquad \forall j, j' \in \mathcal{I}. \tag{13}$$

At iteration r, solving the program yields a solution vector $(\mathbf{d}, \mathbf{e}, \mathbf{x}, \mathbf{y}^+, \mathbf{y}^-, \mathbf{z}, \boldsymbol{\rho}, \mathbf{L^q})$ and the objective value $\mathbf{C}_r$, which is a lower bound for the problem. An upper bound, $\widehat{\mathbf{C}}_r$ can be computed with $\mathbf{C}$ in Eq. 2 using Eq. (1) supported by the recently found $\boldsymbol{\rho}$ input. Once the gap between $\mathbf{C}_r$ and $\widehat{\mathbf{C}}_r$ is satisfactorily low (i.e., approaches to zero), the algorithm terminates and the optimal system cost is $\mathbf{C}^* = \widehat{\mathbf{C}}_r$. A cut created from $j \in \mathcal{I}$ is applicable to all locations as the right-hand side of Eq. (13) activates and deactivates the cut as necessary.

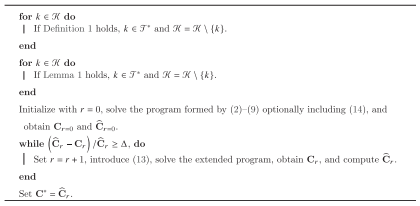
Although Eq. (13) helps to solve the problem to optimality, introducing pre-defined user cuts to the initial program, Eqs. (2)–(9), may potentially hasten the solution process by enforcing Eq. (10) if ρ_j meets an estimated value. To do so, we define $|\mathcal{N}|$ as the number of user cuts to be introduced to each location $j \in \mathcal{I}, \tilde{\rho}_{jn}$ as a constant estimate of ρ_j, ϵ as a small number satisfying $\epsilon < \min_{k \in \mathcal{K}} \lambda_k/\mu$, and $\alpha_{jn} = \{0,1\}$ as a binary decision variable.

$$\rho_j - \tilde{\rho}_{jn} + \epsilon \leqslant \alpha_{jn}, L_j^q \geqslant L_j^q(\tilde{\rho}_{jn}) - \mathbb{M}_2(1 - \alpha_{jn}) \qquad \forall j \in \mathcal{I}, n \in \mathcal{N}. \tag{14}$$

Constraint set Eq. (14) introduces a lower bound to L_j^q, if $\rho_j \geqslant \tilde{\rho}_{jn}$ by enforcing $\alpha_{jn} = 1$. When $\rho_j < \tilde{\rho}_{jn}, \alpha_{jn}$ is free to take the value 0 or 1. Yet, $\alpha_{jn} = 0$ will be preferred since L_j^q is minimized in the objective function. In these constraints, note that L_j^q is a variable, and $L_j^q(\tilde{\rho}_{jn})$ is a function given in Eq. (1). Moreover, these cuts do not guarantee a faster convergence; introducing a set of user cuts may also slow down the convergence due to additional variables and the inclusion of a big number, $\mathbb{M}_2$. The estimation parameter, $\tilde{\rho}_{jn}$, plays a role on the convergence benefit of these cuts.

The algorithm details can be seen in Algorithm 1, and Δ denotes the maximum allowed gap within the bounds. In the solution procedure, two different *gap* concepts arise. First, the solver reports a gap between the lower and upper bounds for the problem with the set of constraints added. If Eq. (10) is not introduced, and Eq. (13) is added iteratively, the objective value provided by the solver is the current lower bound ($\mathbf{C}_r$) and the objective value with re-calculated $\boldsymbol{\rho}$ based on the optimized assignments is the overall upper bound, $\widehat{\mathbf{C}}_r$, for the problem. Hence, $\left(\widehat{\mathbf{C}}_r - \mathbf{C}_r\right)/\widehat{\mathbf{C}}_r$ is the overall gap at iteration r. A total convergence can be achieved when the gap equals to 0, and the optimal solution is denoted by $\mathbf{C}^*$. In the following sections, the gap we refer to is the overall gap at the last iteration.

Algorithm 1. DDNP cutting-plane algorithm.



5. Numerical results

In this section, we first provide the data layout and the design of experiments. Then, we carry out a computational analysis to reveal the capabilities of our approach followed by a case study to point out key system performance metrics. Lastly, we conduct a sensitivity analysis to reveal the impact of key parameters on the total system cost and other related performance metrics.

5.1. Design of experiments

We use a portion of the Chicago, IL Metropolitan area to conduct numerical experiments. As we consider a single DC, we limited the drone service coverage by a bounded area. This area is around 2,600 square miles. We obtained a shape file from the planning and operations language for agent-based regional integrated simulation (POLARIS) tool of the Argonne National Laboratory (Auld et al., 2016). The file contains a complete Chicago Metropolitan area traffic analysis map with a variety of attributes, and the ones considered in this study are 1,961 traffic analysis zones (TAZs), TAZ populations, latitude, and longitude information. Using QGIS 3.4 (QGIS, 2020), first, we reduced the number of TAZs to 1,361 by removing the ones falling out of our pre-determined area. Then, we added centroids to all remaining TAZs. Since considering nearly 7.5 million individual customer vertices would form a computationally challenging problem, and such a large population cannot be realistically served by a single DC, we use the centroids to represent an aggregate demand at a TAZ level and randomly (from a uniform distribution) select a subset of TAZs from 1,361 TAZs to carry out our experiments. The exact problem sizes are provided in the following sections where appropriate. We approximate λ_k by considering TAZ level populations and an estimate daily package demand of 0.01051737 per person as Shavarani et al. (2019) adopted (Kim, 2016; Sudbury and Hutchinson, 2016).

We assume that the DC is located in the center of the area and candidate ABSMs are uniformly distributed in a way that hopping from one to another without battery swap is possible for the given R. For instance, let $R = 10$ minutes (equivalent to 10 miles with a constant 60 mph drone speed), then, $|\mathcal{I}| \approx 40$. The following sec-

tions will provide appropriate values of $|\mathcal{I}|$ and R for different scenarios. We use Euclidean distances to calculate T_{ij}^{δ} with a given drone speed as employing the more accurate Haversine distances make the illustration difficult to interpret.

Based on our interview with Asylon, the average battery swapping time, $1/\mu = 3$ minutes; $C_j^{\phi} = U(60,000, 80,000)$ USD annually (where \$50,000 constitutes to the ABSM and the rest is land use); $C^{\theta} = 2,122$ USD annually [considering 5 year lifetime, \$10,000 drone price, and using the same annual value formulation in] (Shavarani et al., 2019). We also adopt $C^{\delta} = 0.016$ USD per minute assuming \$0.01 per km and a drone speed of 60 mph Shavarani et al. (2019). 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 MILP solver Gurobi 9.0.2 (Gurobi Optimization, LLC, 2020).

5.2. Computational experiments

In this section, we randomly (from a uniform distribution) select 5, 10, 15, 20, 25, and 50 TAZs from 1,361 TAZs as set sizes for $\mathcal{K}$ to test different cases with different solution methods. The total population sizes in these cases are between 30 and 279 thousands, and hence the corresponding total daily units demanded are approximately between 0.2 and 2.9 thousands, which can be thought reasonable for a single DC. We use $|\mathcal{I}| = 40$ and $|\mathcal{I}| = 60$ with uniformly distributed candidate ABSMs and $R = 10$ minutes. Since the analysis of customers within the DC service range is straightforward with a simple cost comparison between drones and trucks, we do not consider these customers and select customers beyond the DC coverage. In the following section, a map illustrating the network will be included to better visualize the problem.

We use three different solution methods to observe their computational efficiencies. For easier referencing, the methods are called: *Cut*, *User cut*, and *Nonlinear*. In Cut method, problem instances are solved by using Eqs. (2)–(9) with iterative addition of Eq. (13). In User cut, we follow Algorithm. 1 by including Eq. (14). For every location $j \in \mathcal{I}$, we use $\tilde{\rho}_{jn} = [0.1, 0.9]$ with step size of 0.1. In Nonlinear, we directly solve Eqs. (2)–(10) without using any additional cut constraints.

Table 3 shows the computational results for a set of instances solved by the tree methods. Here, each instance represents a problem with different customer (aggregated) locations. The reported gap values are the averages for their corresponding instance sizes. For all instances, we set the maximum allowed gap, $\Delta = 5.0 \times 10^{-4}$. For each instance, we limited the computational time by an hour, unless otherwise stated. Hence, the solver stops whichever limitation is reached first. As the table reveals, Cut is generally the best method in terms of computational time and satisfactorily low gap values. We observe that User cut performs the best in small-sized instances, where $|\mathcal{I}| = 40$ and $|\mathcal{K}| = 5$. Further, it also provides lower gap values than Cut in most cases. User cut is also useful in instances first iteration of which consumes all of the allocated time. For example, in five instances where $|\mathcal{I}| = 60$ and $|\mathcal{K}| = 50$ (last row of Table 3), Cut used the allocated one-hour in the first iteration. Therefore, any cutting-planes that would highly reduce the considerably large gap (averaging 2.7×10^{-1}) could not be introduced. User cut, on the other hand, terminated with lower gap values (averaging 7.3×10^{-2} for five instances), although it also consumed the allocated one-hour in the first iteration. Lastly, Nonlinear does not perform well compared to the other methods.

Apart from its better performance in large-scale instances, we can see that User cut also performs better than Cut method in small-sized (e.g., $|\mathcal{I}| = 40$ and $|\mathcal{K}| = 10$) instances, when we lower Δ from 5.0×10^{-4} to 5.0×10^{-8}. We solved several instances and observed that the gap approached to 0 (i.e., 1×10^{-15}) in both methods. In all these instances, User cut performed better in terms of the computational time and required fewer number of iterations to converge. To illustrate our observations, we provide Fig. 2 based on an instance with $|\mathcal{I}| = 40$ and $|\mathcal{K}| = 10$. In this figure, lower and upper bounds for Cut method is shown on the top and for the User cut on the bottom. The notation Gap_r in the middle of dashed lines denotes the overall gap at the end of iteration r. A lower gap at the end of iteration 0 is observed in User cut, while Cut almost catches up with User cut (iteration 1) on its iteration 2. Eventually, User cut converges slightly earlier than Cut. To sum up, user defined cuts can be useful to achieve a low gap in the first iteration (compare 3.54×10^{-1} to 5.23×10^{-2}) and a faster convergence in some cases.

5.3. Case study

In this section, we conduct a case study in the aforementioned Chicago area to show the applicability of our model and share the key performance metrics need to be considered during the decision-making process. We use an instance with the following properties: $|\mathcal{I}| = 40$, $|\mathcal{K}| = 50$, $R = 10$ minutes, and all other parameter values mentioned in Section 5.1 with $\Delta = 5 \times 10^{-8}$. In this instance, there are a total of 277,512 people in 50 TAZs and around 2,919 packages need to be delivered (daily) by drones or trucks. Fig. 3a and b visualize the problem and the solution layouts for the case study, respectively. In the solution figure, black, dark green, red, dark blue, and magenta colored points respectively denote the DC, open (i.e., $x_j = 1$) ABSMs, closed ABSMs, drone cus-

Table 3
Solution statistics for computational experiments.

$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{K}\rvert$	Instances	Average Gap			Average Time (s)		
			Cut	User cut	Nonlinear	Cut	User cut	Nonlinear
40	5	50	3.7 E−5	2.0 E−5	3.4 E−5	15.1	14.0	23.9
	10	25	8.8 E−5	5.9 E−5	2.5 E−2	19.1	26.0	207.1*
	15	10	9.4 E−5	6.5 E−5	7.6 E−2	28.4	74.9	300.0*
	20	5	1.1 E−4	2.2 E−5	11.9 E−2	112.9	464.6	600.0**
	25	5	2.2 E−4	9.1 E−5	11.2 E−2	123.7	314.6	600.0**
	50	5	2.8 E−4	2.8 E−2	5.8 E−2	1236.0	3600.0	3600.0
60	5	50	5.6 E−5	2.4 E−5	1.1 E−2	20.1	30.6	36.9
	10	25	7.3 E−5	6.0 E−5	2.5 E−2	41.2	141.6	249.0*
	15	10	1.3 E−4	1.3 E−4	5.7 E−2	74.6	427.9	600.0**
	20	5	7.5 E−5	7.0 E−2	1.2 E−1	140.2	506.3**	600.0**
	25	5	2.3 E−4	8.7 E−2	1.3 E−1	187.3	600.0**	600.0**
	50	5	2.7 E−1	7.3 E−2	2.1 E−1	3600.0	3600.0	3600.0

* Time is capped by 300 s.
** Time is capped by 600 s.

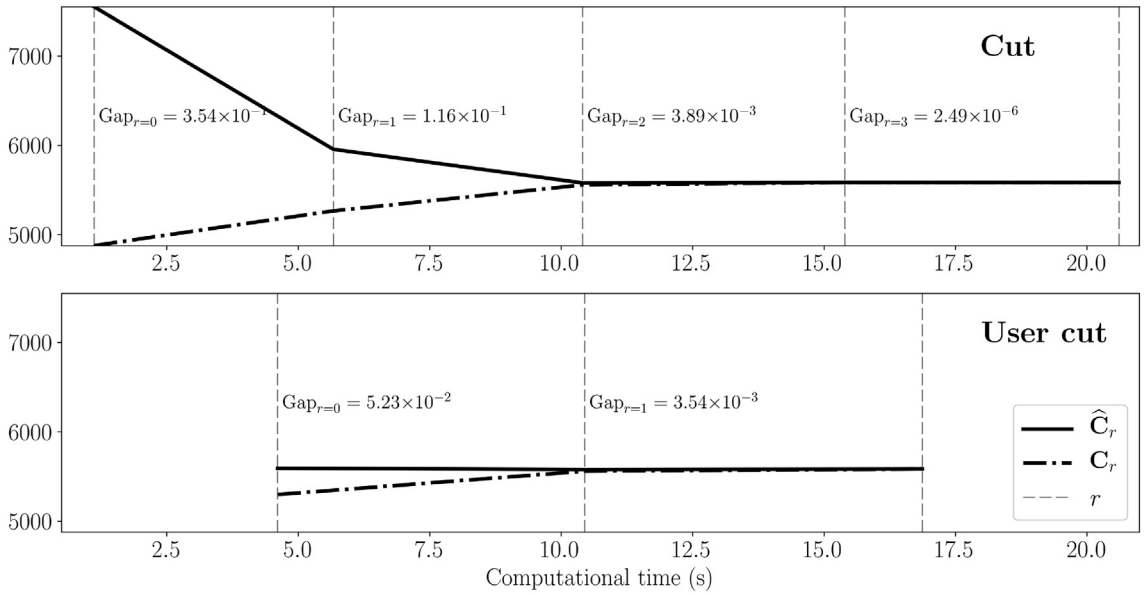


Fig. 2. A convergence example for Cut and User cut methods.

tomers (i.e., $\sum_{i\in\mathcal{I}'_k} y^+_{ikk} = 1$), and truck customers. In this figure, we show R only for the open ABSMs to better visualize the region covered by drones. We also note that the optimal solution for the case is reached in approximately 0.7 hour with 2.2×10^{-16} solution gap.

In Fig. 3b, we observe that there are not any customers within the immediate service range of several ABSMs. This is because these locations serve as hopping stations to extend the drone service to some customers, farther from the DC and are economically better to be served by drones. Although serving more distant customers requires ABSM investment along the path and incurs drone ownership costs, sometimes (as seen in the case), drones can be the optimal delivery mode regardless of these additional cost components. We also observe some customers being served by trucks, although they are under the drone service coverage of an open ABSM. This is due to the queueing effect. Since assigning more drones to those ABSMs will increase the overall drone delivery costs, such customers are served with trucks. Further, another reason is the utilization rate ρ_j limitation at open ABSMs. Since, $\rho_j < 1$ (i.e., demand rate has to be strictly less than the service rate) has to hold, some "within drone service coverage" customers may not be served by drones. In addition to providing key solution metrics, the case study illustrates the role of queueing aspect in the DDNP solution.

We provide Table 4 to share important solution metrics. Out of 50 TAZs, 13 (corresponding to 30% of the population) are served by drones. The table also shows average ρ_j and L_j^q–denoted with a bar–for active ABSM locations. Observing $\bar{\rho}_j = 0.73$, we can suspect that some more customers could be served by some ABSMs to maximize their utilization. However, the additional demand rate introduced into ABSM vertices may not make this possible due to $\rho_j < 1$ restriction. For some customers, whose demand would not cause $\rho_j \geqslant 1$, the nonlinear increase in L_j^q may eliminate the economic feasibility of drones. For these reasons, the optimal solution becomes in the form as presented in Fig. 3b.

Apart from the solution metrics, we also investigated the percent contribution of objective function components to the overall system cost. The component-wise costs were provided in Section 3

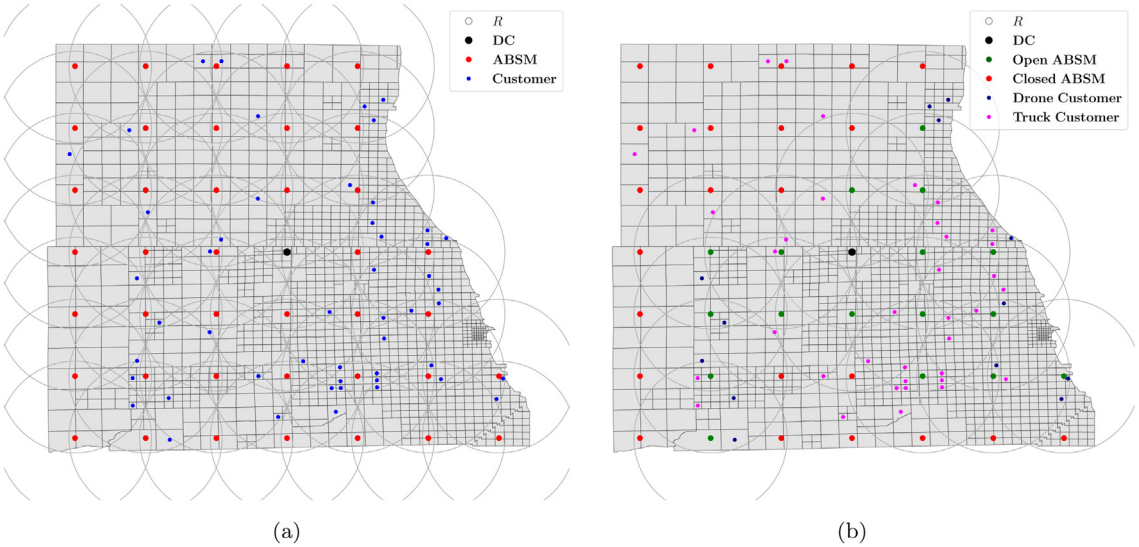


Fig. 3. Case study (a) problem layout and (b) solution visualization.

Table 4
Case study solution metrics.

Number of TAZs served by drones	Number of people served by drones	Number of packages delivered by drones (daily)	Number of ABSMs located	Average number of drones used	$\bar{\rho}_j$	$\bar{L}_j^q$
13	83,125	874	17	91	0.73	0.95

as 1) ABSM, 2) truck delivery, 3) drone delivery, 4) drone ownership, and 5) lead time costs. In this order, these components contribute to $\mathbf{C}^*$ by 9.3, 83.3, 2.7, 1.2, and 3.3 percents. As seen in these percents, lead time costs (related to L_j^q) is a major cost source in the drone delivery system, which highlights the importance of the queueing aspect in this strategic decision-making framework.

5.4. Sensitivity analysis

Since drone delivery and ABSM technologies are in their infancy, decrease in some parameter values can be expected in the future. A sensitivity analysis is conducted to consider the impact of decrease in C^θ, C_j^ϕ, and $1/\mu$ on the DDNP. Hence, we observe system cost and optimal network configuration changes along with a percent decrease in these parameters. We use 10 instances with $|\mathcal{I}| = 40$ and $|\mathcal{K}| = 10$, and consider up to 50% decrease at 5% intervals. Notations such as, $\left[\mathbf{C}^*|C^\theta\right]$, are used to denote the performance metric (system cost in this particular case) given the decrease percent in one of the three parameters (e.g., C^θ). For the sake of brevity, we refrain explicitly describing these notations and note that a decrease in $1/\mu$ corresponds to an increase in μ. Fig. 4 shows the average impact on the five instances with respect to percent decrease in the key parameters.

Based on Fig. 4a, μ and C_j^ϕ are the most critical parameters toward reducing the optimal network cost. We observe that C^θ is not a highly important player in $\mathbf{C}^*$ compared to others. In

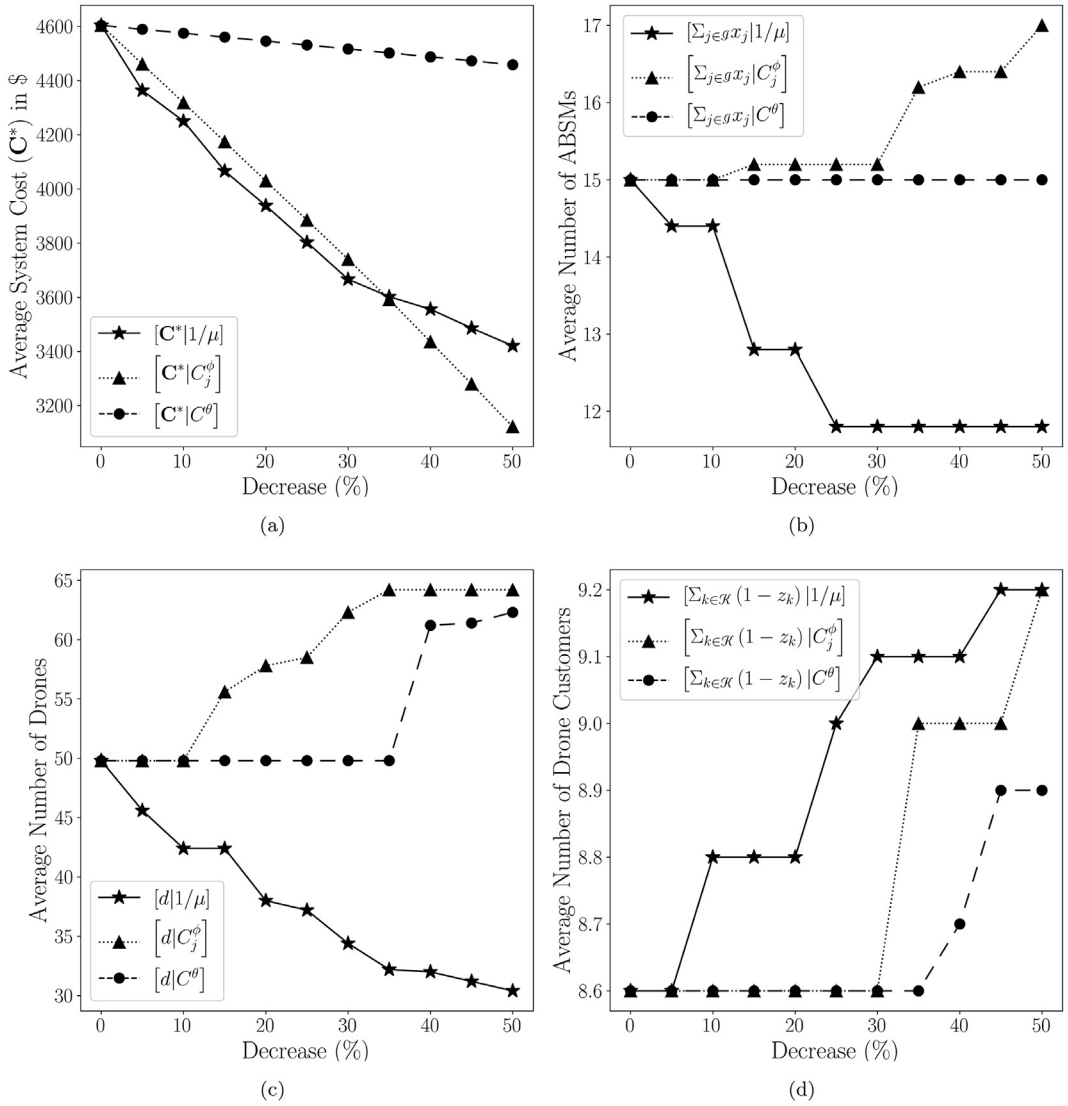


Fig. 4. Impact of percent improvement in key parameters on average (a) system cost, (b) number of ABSMs opened, (c) number of drones, and (d) number of drone customers (at TAZ level).

Fig. 4b, we observe that μ and C_j^{ϕ} significantly impact the network layout for ABSMs. In Fig. 4c, we see how μ and C_j^{ϕ} impact the average number of drones. Further, a large decrease in C^{θ} also increases the average number of drones. Fig. 4d indicates that μ is a critical player in determining the optimal delivery mode. Overall, improvement in ABSMs can make the drone delivery more favorable from many aspects.

In addition to parameter sensitivity, we investigated the benefits of using this model–partitioning the demand locations into truck and drone mode-choices-compared to a sole truck and a sole drone delivery system adoption. We ran 50 instances with $\mathcal{I} = 40$ and $\mathcal{K} = 10$ with $\Delta = 5.0 \times 10^{-4}$. Compared to a sole drone and a sole truck delivery system, our experiments showed that the presented model can reduce the overall system cost by an average of 6.5% and 54%, respectively.

6. Conclusion

In this study, we focused on the current state of the drone technology and identified that their short travel ranges aggravate the realization of a sole drone delivery system. To resolve the range issue, we investigated a drone delivery network with automated battery swapping machines. We employed $M/D/1$ queueing theory to consider the length of the queue at the machines and to prevent a possible bottleneck in the delivery network. Using the length as a performance metric, we prevented the occurrence of a possible bottleneck due to accumulation of drones waiting for battery swap service at certain locations. Then, we formulated an MINLP to locate these machines while minimizing the machine, drone ownership, and value of delivery time costs. To deal with the nonlinearity, we proved the convexity of the waiting line length in utilization rate and developed a derivative-supported cutting-plane algorithm. Using the algorithm, we iteratively solved the mixed-integer program by introducing cut constraints. The algorithm was proven to converge to optimality.

To test the performance of our solution method, validate the applicability of our program, and investigate how the key parameters impact the future adoption of drones for delivery, we conducted numerical experiments. We found that our algorithm coupled with a commercially available solver performs well for a problem with 25 aggregate customer locations and 40 candidate machine locations. Our case study focusing on the Chicago Metropolitan area has also proven the applicability of our program by solving the problem with 0.03% optimality gap in 4.6 h. In the sensitivity analysis, we found that battery swapping time followed by its cost are the key parameters impacting the system cost and numbers of required drones and swapping machines in the system.

Our model has several limitations. It does not consider the battery cost as doing so would result in solving a more difficult problem. Although batteries are not as costly as drones, each ABSM has to carry a large battery inventory, which may alter the decisions. As an extension, we will develop another model to incorporate battery costs and consider situations where fully-charged batteries are not available for swap operations at some time. Considering such situations could challenge the cost efficiency of drones by revealing a better cost schematic. Furthermore, unavailability of drones and disruptions at ABSMs are not considered in this work. Such aspects can be more suitably modeled in a simulation environment.

This study defines a high level problem that does not consider many operational details. For instance, autonomous delivery drones can leave an ABSM queue due to long queue wait times and time-and-weather-dependent travel times can be used to better route drones in an operational decision-making framework. Even at a strategic level, the problem is quite difficult to *optimally* solve. Hence, heuristic solutions which will be addressed in a future study should be developed to diminish the computational complexity.

CRediT authorship contribution statement

Taner Cokyasar: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Wenquan Dong:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Mingzhou Jin:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **İsmail Ömer Verbas:** Conceptualization, Writing - original draft.

Acknowledgements

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 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

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 52, 965–981. https://doi.org/10.1287/trsc.2017.0791.

Amazon, Inc, 2020. Amazon prime air. Available at URL:http://www.amazon.com/primeair, accessed on Sep. 29, 2020..

Asylon, Inc, 2020. An aerial infrastructure company. Available at URL:https://www.flyasylon.com, accessed on Sep. 29, 2020..

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

Batta, R., 1989. Technical note-the stochastic queue median over a finite discrete set. Oper. Res. 37, 648–652. https://doi.org/10.1287/opre.37.4.648.

Batta, R., Berman, O., 1989. A location model for a facility operating as an M/G/k queue. Networks 19, 717–728. https://doi.org/10.1002/net.3230190609.

Berman, O., Drezner, Z., 2007. The multiple server location problem. J. Oper. Res. Soc. 58, 91–99. https://doi.org/10.1057/palgrave.jors.2602126.

Berman, O., Larson, R.C., Chiu, S.S., 1985. Optimal server location on a network operating as an M/G/1 queue. Oper. Res. 33, 746–771. https://doi.org/10.1287/opre.33.4.746.

Berman, O., Larson, R.C., Parkan, C., 1987. The stochastic queue p-median problem. Transport. Sci. 21, 207–216. https://doi.org/10.1287/trsc.21.3.207.

Bouman, P., Agatz, N., Schmidt, M., 2018. Dynamic programming approaches for the traveling salesman problem with drone. Networks 72, 528–542. https://doi.org/10.1002/net.21864.

Buckley, J., 2006. Air Power in the Age of Total War. Routledge, London, United Kingdom, ISBN 978-0-253-21324-2.

Chiu, S.S., Larson, R.C., 1985. Locating an n-server facility in a stochastic environment. Comput. Oper. Res. 12, 509–516. https://doi.org/10.1016/0305-0548(85)90050-4.

Clarke, R., 2014. Understanding the drone epidemic. Comput. Law Secur. Rev. 30, 230–246. https://doi.org/10.1016/j.clsr.2014.03.002.

Cokyasar, T., Dong, W., Jin, M., 2019. Network optimization for hybrid last-mile delivery with trucks and drones. In: IISE Annual Conference and Expo 2019,

Institute of Industrial and Systems Engineers (IISE), pp. 1250–1255. ISBN: 9781713814092..
Curlander, J.C., Gilboa-Amir, A., Kisser, L.M., Koch, R.A., Welsh, R.D., 2017. Multi-level fulfillment center for unmanned aerial vehicles. US Patent 9,777,502..
de Freitas, J.C., Penna, P.H.V., 2020. A variable neighborhood search for flying sidekick traveling salesman problem. Int. T. Oper. Res. 27, 267–290. https://doi.org/10.1111/itor.12671.
Dell'Amico, M., Montemanni, R., Novellani, S., 2019. Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem. Optim. Lett. 2019, 1–32. https://doi.org/10.1007/s11590-019-01492-z.
FAA, 2016. Summary of small unmanned aircraft rule (Part 107). Federal Aviation Administration, Washington, DC. Available at URL: https://www.faa.gov/uas/media/Part_107_Summary.pdf, accessed on Sep. 29, 2020..
Gurobi Optimization, LLC, 2020. Gurobi optimizer reference manual. Available at URL:http://www.gurobi.com, accessed on Sep. 29, 2020..
Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. Transp. Res. Part C Emerg. Technol. 86, 597–621. https://doi.org/10.1016/j.trc.2017.11.015.
Hamaguchi, T., Nakade, K., et al., 2010. Optimal location of facilities on a network in which each facility is operating as an M/G/1 queue. J. Serv. Sci. Manage. 3, 287–297. https://doi.org/10.4236/jssm.2010.33036.
Holland, C., Levis, J., Nuggehalli, R., Santilli, B., Winters, J., 2017. UPS optimizes delivery routes. Interfaces 47, 8–23. https://doi.org/10.1287/inte.2016.0875.
Hong, I., Kuby, M., Murray, A.T., 2018. A range-restricted recharging station coverage model for drone delivery service planning. Transp. Res. Part C Emerg. Technol. 90, 198–212. https://doi.org/10.1016/j.trc.2018.02.017.
Huang, H., Savkin, A.V., Huang, C., 2020a. A new parcel delivery system with drones and a public train. J. Intell. Robot. Syst. 2020, 1–14. https://doi.org/10.1007/s10846-020-01223-y.
Huang, H., Savkin, A.V., Huang, C., 2020b. Round trip routing for energy-efficient drone delivery based on a public transportation network. IEEE Trans. Transp. Electr. 6, 1368–1376. https://doi.org/10.1109/TTE.2020.3011682.
Kelley, Jr, J.E., 1960. The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. 8, 703–712. doi: 10.1137/0108053..
Kim, E., 2016. The most staggering part about Amazon's upcoming drone delivery service. Available at URL:https://www.businessinsider.com/cost-savings-from-amazon-drone-deliveries-2016-6, accessed on Sep. 29, 2020..
Kim, S.J., Lim, G.J., 2018. Drone-aided border surveillance with an electrification line battery charging system. J. Intell. Robot. Syst. 92, 657–670. https://doi.org/10.1007/s10846-017-0767-3.
Kim, S.J., Lim, G.J., Cho, J., 2018. Drone flight scheduling under uncertainty on battery duration and air temperature. Comput. Ind. Eng. 117, 291–302. https://doi.org/10.1016/j.cie.2018.02.005.
Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M., Brunese, P.A., 2019. Multiple traveling salesman problem with drones: mathematical model and heuristic approach. Comput. Ind. Eng. 129, 14–30. https://doi.org/10.1016/j.cie.2019.01.020.
Kleinrock, L., 1975. Queueing systems. Volume I: Theory. John Wiley & Sons, New York, NY. ISBN: 978-0-471-49110-1..
Larson, R.C., 1974. A hypercube queuing model for facility location and redistricting in urban emergency services. Comput. Oper. Res. 1, 67–95. https://doi.org/10.1016/0305-0548(74)90076-8.
Mak, H.Y., Rong, Y., Shen, Z.J.M., 2013. Infrastructure planning for electric vehicles with battery swapping. Manage. Sci. 59, 1557–1575. https://doi.org/10.1287/mnsc.1120.1672.
McKenna, A., 2016. The public acceptance challenge and its implications for the developing civil drone industry. In: Custors, B. (Ed.), The Future of Drone Use. T.M.C. Asser Press - Springer, The Hague, The Netherlands, pp. 353–369. ISBN: 978-9-462-65131-9..
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transp. Res. Part C Emerg. Technol. 54, 86–109. https://doi.org/10.1016/j.trc.2015.03.005.
Murray, C.C., Raj, R., 2020. The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones. Transp. Res. Part C Emerg. Technol. 110, 368–398. https://doi.org/10.1016/j.trc.2019.11.003.
Park, S., Zhang, L., Chakraborty, S., 2017. Battery assignment and scheduling for drone delivery businesses. In: 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), IEEE, pp. 1–6. doi: 10.1109/ISLPED.2017.8009165..
QGIS, 2020. Qgis user guide. Available at URL:https://docs.qgis.org, accessed on Sep. 29, 2020..
Scott, J.E., Scott, C.H., 2019. Models for drone delivery of medications and other healthcare items. In: Unmanned Aerial Vehicles: Breakthroughs in Research and Practice. IGI Global, pp. 376–392. doi: 10.4018/978-1-5225-8365-3.ch016..
Shavarani, S.M., Mosallaeipour, S., Golabi, M., Izbirak, G., 2019. A congested capacitated multi-level fuzzy facility location problem: an efficient drone delivery system. Comput. Oper. Res. 108, 57–68. https://doi.org/10.1016/j.cor.2019.04.001.
Snyder, L.V., 2006. Facility location under uncertainty: a review. IIE Trans. 38, 547–564. https://doi.org/10.1080/07408170500216480.
Stewart, W.J., 2009. Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling. Princeton University Press, Princeton, NJ, ISBN 978-0-691-14062-9.
Sudbury, A., Hutchinson, E., 2016. A cost analysis of amazon prime air (drone delivery). J. Econ. Educ. 16, 1–12. ISSN: 2688-5956.
Wang, Q., Batta, R., Rump, C.M., 2002. Algorithms for a facility location problem with stochastic customer demand and immobile servers. Ann. Oper. Res. 111, 17–34. https://doi.org/10.1023/A:1020961732667.
Winston, W.L., 2004. Operations Research: Applications and Algorithms. Brooks/Cole–Thomson Learning, Belmont, CA. ISBN: 0-534-38058-1.
Xie, F., Liu, C., Li, S., Lin, Z., Huang, Y., 2018. Long-term strategic planning of inter-city fast charging infrastructure for battery electric vehicles. Transp. Res. E Logist. Transp. Rev. 109, 261–276. https://doi.org/10.1016/j.tre.2017.11.014.