Computers and Operations Research 113 (2020) 104802

Contents lists available at ScienceDirect

Computers and Operations Research

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

Multi-visit drone routing problem

Stefan Poikonen[a,*], Bruce Golden[b]

[a] University of Colorado Denver - Business School, United States
[b] University of Maryland - R.H. Smith School of Business, United States

ARTICLE INFO

Article history:
Received 18 September 2018
Revised 28 July 2019
Accepted 13 September 2019
Available online 17 September 2019

MSC:
Routing truck/drone tandems where drone may carry multiple packages
Energy constraints

Keywords:
Vehicle routing
Traveling salesman
Drones
Heuristics

ABSTRACT

The k-Multi-visit Drone Routing Problem (k-MVDRP) considers a tandem between a truck and k drones. (When $k = 1$, the problem is called the Multi-Visit Drone Routing Problem.) Each drone is capable of launching from the truck with one or more packages to deliver to customers. Each drone may return to the truck to swap/recharge batteries, pick up a new set of packages, and launch again to customer locations. Unlike many papers in the current literature, the model not only allows for a drone to carry multiple heterogeneous packages but also allows the specification of a drone energy drain function that takes into account each package weight, and it decouples the set of launch locations from the set of customer locations. This paper proposes a flexible heuristic solution. Computational experiments and sensitivity analyses are also conducted using physical parameters for drones that are consistent with recent research.

1. Introduction

In a 2013 interview, Amazon CEO Jeff Bezos suggested Amazon would eventually be capable of reducing costs by delivering packages up to five pounds (2.3 kg) within 30 min of order placement via drone (Bezos, 2013). Later, Amazon publicly announced in an April 18, 2018 letter to shareholders that the Amazon Prime service, which offers free two-day shipping of millions of Amazon products, had eclipsed 100 million subscribers and that year 2017 saw the largest subscriber growth in the history of Amazon Prime (Spangler, 2018). Naturally, the combination of potentially offering rapid delivery of online orders with cost savings to a massive and growing market caught the attention of many in the operations research community.

Amazon is not alone in their pursuit of drones for delivery. FedEx, UPS, Posti, Google, Russian Post, and DPD all have reportedly been testing drones for use in delivery (Cary and Bose, 2017; Rey, 2016; Reuters, 2015, 2018; UPS, 2017; Wing, 2018), as drone delivery may offer cost savings, reduced emissions, and lower delivery times. If Amazon's Prime Air project (Air, 2018), Google's Project Wing project (Wing, 2018), and similar ventures come to fruition, then we must consider models that are well-adapted for a drone-specific context.

Some papers, including Murray and Chu (2015) and Ulmer and Thomas (2018), have studied a warehouse direct to consumer model of drone delivery. In one of two models studied by Murray and Chu (2015), customers within range of a warehouse may receive packages by a drone, whereas further away customers may receive packages by truck. Ulmer and Thomas (2018) consider a same-day delivery scenario, where trucks and drones are dispatched throughout the day as orders are placed. They study the effect of different policies based on geographic districts and distance thresholds for allocating packages to either a truck or drone for delivery.

Due to range limitations related to the finite battery capacity or communication range (Lillian, 2018) of the drone and the desire to limit the number of warehouses due to economies of scale, other papers have considered hybrid truck-and-drone models of delivery, which mitigate the impact of the limited range of the drone. In Dayarian et al. (2018), the drone flies from the warehouse to resupply a truck which is out on a delivery route. Campbell et al. (2017) use continuous approximation models to assess the economic impact of truck-and-drone hybrid models over a variety of model parameters and customer densities. Carlsson and Song (2017) study continuous approximation on the Horse Fly Problem, where the truck serves as a mobile depot for the drone, and conclude the efficiency gain is

* Corresponding author.
E-mail addresses: stefan.poikonen@ucdenver.edu (S. Poikonen), bgolden@rhsmith.umd.edu (B. Golden).

https://doi.org/10.1016/j.cor.2019.104802
0305-0548/

proportional to the squareroot of the speed ratio between drone and truck. Other papers, including Ha et al. (2018), Agatz et al. (2018), Poikonen et al. (2017), Wang et al. (2017), Poikonen et al. (2019), and Murray and Chu (2015), that consider hybrid truck-and-drone delivery typically allow the truck to act as a mobile depot and battery swap station for the drone, which then delivers the package directly to a customer location.

Many of these papers use some (if not most) of the following simplifying assumptions.

1. All packages are homogeneous.
2. The drone is capable of carrying a single package at a time.
3. The battery life of a drone is a fixed amount of time, not dependent on the weight of the package(s) it is carrying.
4. The speed of the drone is fixed and is not a decision variable capable of being optimized.
5. The set of allowable locations to launch/retrieve a drone from the truck is identical to the set of customer locations.

These assumptions may be appropriate in a variety of circumstances. However, in other cases, it may be more appropriate to relax some or all of these assumptions.

With respect to simplifying assumption (1), there are many contexts where the weights of packages may vary significantly. With respect to (2), the vast majority (86%) of packages delivered by Amazon are less than 2.3 kg as of 2013 (Bezos, 2013). Yet, there already exist commercial drones that can carry several multiples of that payload, including the FreeFly Alta 8, which is capable of carrying payloads of up to 18 kg (Flynt, 2018). The largest manufacturer of drones in the world (Borak, 2018), Shenzhen-based DaJiang Innovations (DJI), also produces drones capable of carrying several consumer parcels. DJI sells the DJI Spreading Wings S900 for approximately $2000 (as of September 2018), which is capable of carrying a payload of 8.2 kg (DJI, 2018b). Griff Aviation is designing vertical take-off, multi-purpose, autonomous drones. The Griff 300 is designed with a goal of allowing a 300 kg payload; the Griff 800 is designed with a goal of allowing up to 800 kg payloads (Flynt, 2018). Boeing has also tested prototype vertical take-off cargo drones, which will eventually be able to carry payloads ranging from 250 to 500 pounds while traveling 60–70 miles per hour at a few hundred feet of altitude, according to HorizonX division leader Peter Kunz (Boeing, 2018; Davies, 2018). Thus, there already exist drones with the technical capabilities of carrying the weight of multiple typical consumer packages simultaneously. (We note, however, that heterogeneous package sets introduce a number of technical and design challenges, including issues related to package standardization and off-loading, which are beyond the scope of this paper.)

With respect to (3), just like many other aircraft, drones typically require more power while carrying heavier payloads (DJI, 2018a; Flynt, 2018; Stolaroff and Joshuah, 2018). With respect to (4), it should be noted that the speed of a drone that minimizes energy expenditure per unit distance is frequently slower than the maximum speed of the drone, as drag is a superlinear function of speed. Moreover, these energy- or time-minimizing speeds are dependent on the payload carried by the drone at any given time (Franco and Buttazzo, 2015). With respect to (5), there is no reason to assume that the ideal locations to launch/retrieve a drone are necessarily customer locations. Moreover, in practice, not all customer locations are suitable locations for launching/retrieving the drone from a truck.

Therefore, we will propose a new truck-and-drone delivery model, which relaxes many of the assumptions above. Additionally, we provide thoroughly commented code in the hope that it may be useful to others in the research community.

In Section 2, we formally define the Multi-visit Drone Routing Problem. In Section 3, we describe our heuristic solution method. Section 4 contains theoretical results. In Section 5, we describe our computational experiments and analyze the results. Section 6 offers insights and directions for future research.

2. *k*-Multi-Visit Drone Routing Problem: definition

2.1. Description

The *k*-Multi-visit Drone Routing Problem (k-MVDRP) is a problem that seeks to deliver a set of packages to a set of customer locations in a manner that minimizes route completion time. In k-MVDRP, a truck and k drones start at a predefined warehouse. The truck acts as a mobile depot and recharging platform for the drones. The drones may launch from the truck with one or more packages, deliver these packages to their designated destinations, then return to the truck for recharging and to pick up additional packages. Drone flights are subject to additional restrictions that will be detailed in Section 2.3. The route is considered complete when all customer packages have been delivered and when the truck and all drones have returned to the origin warehouse. The special case where $k = 1$ is simply called MVDRP.

2.2. Problem input parameters

In k-MVDRP, C is the set of customer locations that require a package be delivered. For each customer location $c_i \in C$, we use w_i to denote the corresponding weight of the package demanded at c_i. The set V contains all locations where a drone is allowed to launch from or return to the truck. By convention, the warehouse location is denoted $v_0 \in V$.

The truck and drone(s) have different travel times between pairs of locations. For any pair of locations $v_i, v_j \in V$, the travel time for the truck is given by $tt(v_i, v_j)$. For any pair of locations $v_i, v_j \in V \cup C$, the travel time for a drone is given by $td(v_i, v_j)$. We assume the triangle inequality applies for tt and td.

The energy capacity of the battery for each drone is *EMAX*. It is assumed that for any pair of locations $v_i, v_j \in V \cup C$ and for any set of packages with combined weight w, the energy drain per unit time for a drone carrying these packages is $e(v_i, v_j, w)$. We only impose the restriction that the addition of an extra package to the drone's payload can never reduce the energy dissipation rate. The constant *HOV* is the energy dissipation rate (per unit time) of a drone when it is hovering without any payload. This may occur when a drone arrives at a rendezvous location before the truck and must wait for the truck. The constant *LaunchPenalty* is an overhead time penalty applied anytime the truck stops to launch one or more drones. Table 1 summarizes k-MVDRP input parameters.

2.3. Restrictions on operations

Let us define an *operation* as a set of actions beginning with the truck and all drones at a launch location $v_i \in V$ and terminating with the truck and all drones located at $v_j \in V$. At v_i, any of the k drones may launch from the truck, visit one or more customers, then rendezvous with the truck at v_j. (Note: It is allowed, but not required, that $v_i = v_j$.) A k-MVDRP operation requires that after any drones launch from v_i, the truck must travel directly from v_i to v_j without stopping or launching additional drones. This is similar to an assumption made in the Horse Fly Problem of Carlsson and Song (2017), which has the benefit of producing drone flights whose (non-)feasibility are independent of one another, which reduces computational complexity. (However, the inability of the truck to service customers while any drone is airborne is a clear drawback.) Fig. 1 contains two operations. The operation at left is not allowed, because it contains a drone launch in

Table 1
Summary of k-MVDRP model parameters.

Name	Description
$C = \{c_1, c_2, \ldots, c_{\|C\|}\}$	Set of customers demanding delivery.
$W = \{w_1, w_2, \ldots, w_{\|C\|}\}$	Set of package weights demanded by customers.
$V = \{v_1, v_2, \ldots, v_{\|V\|}\}$	Set of locations where launching/landing drone is allowed.
k	Number of drones. $k \in \mathbb{N}$.
$EMAX$	Energy capacity of each drone. $EMAX \in [0, \infty)$.
tt	Travel times for the truck. $tt(v_i, v_j) \geq 0, \forall v_i, v_j \in V$.
td	Travel times for the drones. $td(v_i, v_j) \geq 0, \forall v_i, v_j \in V \cup C$.
HOV	A constant that describes the energy drain per unit time when a drone hovers, waiting for the truck at a rendezvous location.
$LaunchPenalty$	A constant time penalty imposed on any operation that includes a drone launch.
$e(v_i, v_j, w)$	Energy drain per unit time of a drone traveling from v_i to v_j, carrying packages with total weight of w.

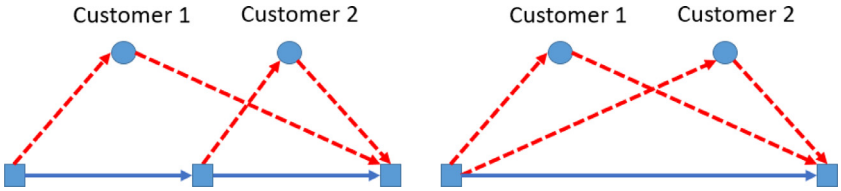

Fig. 1. Blue squares: feasible launch/landing locations for the drone. Blue circles: customer locations. Dashed line segments: drone flights. Blue solid line segments: path of truck. At left, an example of an operation that is not allowed in k-MVDRP. At right, an example of an operation that is allowed, because all drone launches occur at a single location and all drone landings occur at a single location. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the middle of an operation. The operation at right is allowed in k-MVDRP, because all drone launches/landings occur at the start/end points of the operation.

Each drone flight (sortie) within an operation is allowed only when it is *energy feasible*. A drone sortie is considered energy feasible if it expends less than $EMAX$ energy. Formally, if a drone sortie launches from location $v_i \in V$, flies (in order) to customers $c_{s_1}, c_{s_2}, \ldots, c_{s_n}$, then returns to the truck at $v_j \in V$, then the sortie is energy feasible if and only if:

$$\begin{aligned} & e(v_i, c_{s_1}, w_{sum}) * td(v_i, c_{s_1}) + \\ & \sum_{j=2}^{n} e(c_{s_{j-1}}, c_{s_j}, w_{sum} - \sum_{a=1}^{j} w_{s_a}) * td(c_{s_{j-1}}, c_{s_j}) + \\ & e(c_{s_n}, v_j, 0) * td(c_{s_n}, v_j) + \\ & HOV * max(0, tt(v_i, v_j) - droneTime) \\ & \leq EMAX, \end{aligned}$$

where $w_{sum} = \sum_{i \in \{s_1, s_2, \ldots, s_n\}} w_i$ is the combined weight of all packages delivered in the sortie, and where $droneTime = td(v_i, c_{s_1}) + \sum_{a=1}^{j} td(c_{s_{j-1}}, s_{i_j}) + td(s_{i_n}, v_j)$. An operation is infeasible if any constituent drone flight is energy infeasible.

2.4. Formulation

We may formally define k-MVDRP as the following integer linear program.

$$\texttt{Minimize:} \sum_{o \in O} t_o * x_o \quad (C0)$$

`subject to:`

$$t_o \geq truckTime(o), \forall o \in O \quad (C1)$$

$$t_o \geq droneTime_i(o), \forall o \in O, i \in \{1, 2, \ldots, k\} \quad (C2)$$

$$cover(o, j) * x_o \geq 1, \forall j \in \{1, 2, \ldots, |C|\} \quad (C3)$$

$$\sum_{o \in O} rp(o, l) * x(o) = \sum_{o \in O} lp(o, l) * x(o), \forall l \in V \quad (C4)$$

`subtour elimination constraints` (C5)

$$x_o \in \{0, 1\}, \forall o \in O. \quad (C6)$$

In the above formulation, O is the set of all feasible k-MVDRP operations. For an operation $o \in O$, decision variable $x_o = 1$ if the operation is selected as part of the solution. $x_o = 0$, otherwise. The formulation contains several precomputed values. $truckTime(o)$ is the travel time for the truck from the launch point to the retrieval point of operation $o \in O$. $droneTime_i(o)$ is the time required for drone number i to fly from the launch point of o, visit all customers that it has been assigned as part of the operation (in order), then return to the retrieval point of the operation. If drone number i has not been assigned to visit any customers as part of operation o, then $droneTime_i(o) = 0$. $cover(o, j)$ has value 1 only if operation o services customer c_j. Otherwise, $cover(o, j) = 0$. $lp(o, l) = 1$ whenever operation o begins at location l; $rp(o, l) = 1$ whenever operation o terminates at location l. $lp(o, l)$ and $rp(o, l)$ take value 0 otherwise.

Objective (C0) states that we seek to minimize the total route completion time, which is the sum of individual operation times. Constraints (C1) ensure that the time required for an operation is at least as large as the time required for the truck to drive from the launch point to retrieval point of the operation. Constraints (C2) ensure that the time required for an operation is at least as long as the longest drone flight duration within the operation. Constraints (C3) ensure that every customer is serviced (covered). Constraints (C4) are flow constraints, which ensure a closed tour. Constraints (C5) are generic constraints to ensure the truck tour is not disjoint. Constraints (C6) ensure that the selection of any operation is a binary variable.

3. Heuristic approach: route, transform, shortest path

The preceding formulation may quickly become intractable, as the number of feasible drone operations may be extremely large. Thus, we introduce a heuristic solution approach called Route, Transform, Shortest Path (RTS). RTS has three major phases.

3.1. Phase 1: route

Compute the optimal (or heuristic) solution to the standard traveling salesman problem on the set of customer locations and the depot, $C \cup \{v_0\}$, using the drone travel times, td, as the measure

of time between any pair of locations. Let us denote the result as:

$$VisitOrder = [p_0 = v_0, p_1, p_2, \ldots, p_{|C|}, p_{|C|+1} = v_0].$$

The first customer location to be visited is p_1, the second customer location to be visited is p_2, and generally p_i is the ith customer location to be visited.

RTS restricts our search to solutions that have the following properties. (1) If $i < j$, then p_j may not be serviced by an operation that terminates before p_i is serviced. (2) If $i < j$ and p_i and p_j are to be serviced on the same drone sortie, then p_i must be serviced before p_j. These properties imply that customers must be visited in the same order as *VisitOrder*, with possible exceptions when a pair of customers are serviced in the same operation by different drones.

3.2. Phase 2: transform

Let (i, j) represent the condition where truck and all drones are present at physical location v_i and where the first j customer locations (i.e., $p_1, p_2, \ldots, p_j$) have already been serviced.

We construct graph $G' = (V', E)$. For each $i \in V$ and for each $j \in \{0, 1, 2, \ldots, |C|\}$, there exists a vertex $v'(i, j) \in V'$.

Suppose $v'(i1, j1), v'(i2, j2) \in V'$, and let $edge(i1, j1, i2, j2)$ be an edge connecting these vertices. $edge(i1, j1, i2, j2)$ corresponds to an operation that begins at physical location v_{i1} with the first $j1$ customer locations serviced, and terminates at physical location v_{i2} with the first $j2$ customer locations serviced. $edge(i1, j1, i2, j2)$ has a corresponding cost, $cost(i1, j1, i2, j2)$, which is the time required to complete the corresponding operation. We may write:

$$cost(i1, j1, i2, j2) = max(tt(v_{i1}, v_{i2}), max_{a \in \{1,2,\ldots,k\}} droneTime_a(o)),$$

where $droneTime_a(o)$ is the flight time of drone number a within the operation o that corresponds to tuple $(i1, j1, i2, j2)$. $droneTime_a(o)$ is dependent on which customers within the operation are to be visited by drone number a.

We use a heuristic approach to assign customer locations to drones. In the *block assignments* scheme, we assign the first $\lceil (j_2 - j_1)/k \rceil$ customer locations of an operation to the first drone. The next $\lceil (j_2 - j_1)/k \rceil$ are assigned to the second drone, and so on until all customer locations $p_{j_1+1}, p_{j_1+2}, \ldots p_{j_2}$ are assigned. If a drone is assigned customer locations p_s and p_t in the same drone operation and $s < t$, then the block assignment scheme forces the drone to visit p_s before p_t.

In Fig. 2, we display an example of the block assignment scheme applied to $edge(11, 4, 7, 10)$ with three drones. In this example, $j2 = 10$, $j1 = 4$, and $k = 3$, so the first drone must deliver to p_5 and p_6, the second drone must deliver to p_7 and p_8, and the third drone must deliver to p_9 and p_{10}.

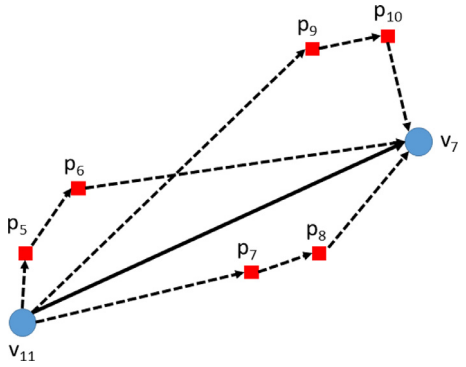

Fig. 2. In this example, we display the block assignment partitioning scheme for $edge(11, 4, 7, 10)$ with $k = 3$ drones available. The dashed lines indicate the flight paths of the three drones. The solid line indicates the movement of the truck. Each drone is assigned a consecutive *block* of customer locations to visit. In this case, each block has size two.

We remind the reader that $VisitOrder = [p_0, p_1, \ldots, p_{|C|+1}]$ is determined in the "Route" stage by solving a TSP using travel times defined by td. Therefore, we naturally expect that consecutive customer locations p_i and p_{i+1} will tend to be located near one another. This is the intuition underlying the block assignment scheme: consecutive blocks of packages are typically located near one another.

In general, the block partitioning scheme is not always the optimal assignment of drones within a single operation. However, in the special case when $k = 1$, the block partitioning makes the only allowable assignment, where all customers are assigned to the same drone, which implies it is the optimal assignment. In the case where $j2 - j1 \leq k$, then each drone will visit only a single customer location, which is optimal due to the triangle inequality.

If the energy expended by each drone under the block partitioning scheme for $edge(i1, j1, i2, j2)$ does not exceed $EMAX$, then the assignment is feasible and $edge(i1, j1, i2, j2) \in E$. Otherwise, $edge(i1, j1, i2, j2) \notin E$.

3.3. Phase 3: shortest path

Phase 3 computes a shortest path over the graph $G = (V', E)$ from vertex $v'_{0,0}$ to $v'_{0,|C|}$. Recall that v_0 is the warehouse location. Thus, Phase 3 may be interpreted as computing a shortest path from "warehouse location (v_0) in a state where 0 packages have been delivered" to "warehouse location (v_0) in a state where all $|C|$ packages have been delivered" using only operations that obey the ordering dictated by *VisitOrder* and do not violate drone energy constraints. The resulting path, denoted $ROUTE(VisitOrder)$, is our heuristic solution to the problem with corresponding objective value denoted $VAL(VisitOrder)$.

In Fig. 3, we display an example MVDRP solution. At left, we visualize the solution path, traced through the graph G'. At right, we show the physical solution path.

4. Theoretical results

Theorem 1. *Among feasible solutions to MVDRP that visit customer locations in the order dictated by VisitOrder, ROUTE(VisitOrder) is the one with the lowest completion time.*

Proof. Recall that MVDRP is the case where $k = 1$. We first observe that any feasible solution to MVDRP can be characterized as a sequence of feasible operations, where the end point of the previous operation is the start point of the next operation. An operation that obeys *VisitOrder* must begin at some location $v_{i1} \in V$, terminate at some location $v_{i2} \in V$, and service some consecutive set of customer locations $p_{j1+1}, p_{j1+2}, \ldots, p_{j2}$. (In the case $j1 = j2$, the customer set is empty and the operation may be viewed as a simple truck relocation from v_{i1} to v_{i2}.)

Observe that in G', for any arbitrary vertex $v'(i1, j1)$, we are allowed to traverse to any other vertex $v'(i2, j2)$ that can be reached by an edge representing a feasible operation that obeys *VisitOrder*. Any two states that can be connected by an operation obeying *VisitOrder* are linked by a directed edge in G'.

Thus, any feasible sequence of operations that obey *VisitOrder* is a set of connected directed edges across G'. A solution for MVDRP must begin at the warehouse v_0 in a state of 0 customers serviced and must terminate at the warehouse with all $|C|$ customers serviced. Thus, any solution to MVDRP can be represented as a path from $v'_{0,0}$ to $v'_{0,|C|}$ on graph G'.

The objective value for an MVDRP solution is simply the sum of the durations of the constituent operations. The cost of each edge

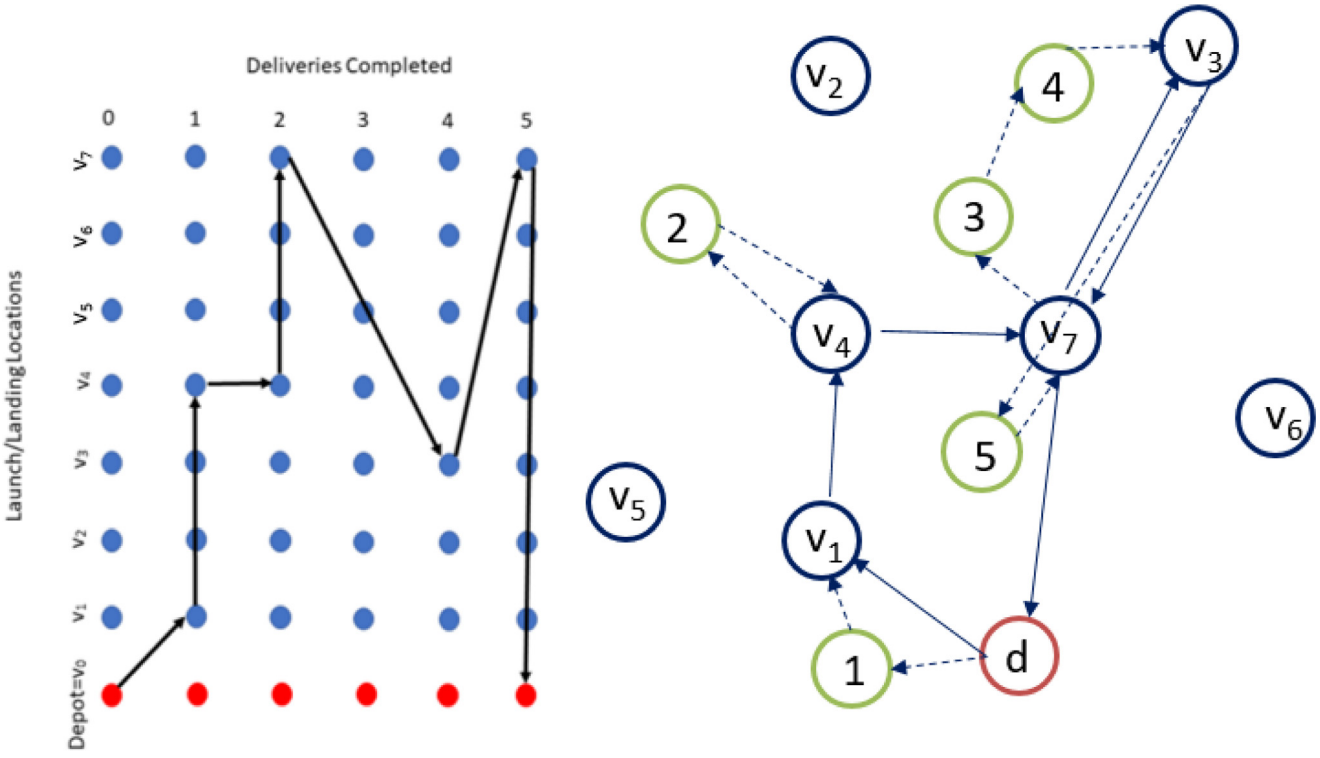

Fig. 3. At left, each row represents a different physical location: $v_0, v_1, \ldots, v_7$. Each column is a different state: 0 deliveries completed, 1 delivery completed, ..., 5 deliveries completed. The grid of vertices is the set V'. Black arrows connecting two vertices indicate that the corresponding operation was chosen as part of the solution. At right, we see the solution traced out in 2-D space. Green circles are customer locations with numeric labels corresponding to *VisitOrder*. The blue circles are launch/landing locations in V. The warehouse or depot is labeled d. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in G' is the duration of the associated operation. Thus, the cost of a path in G', which is the sum of costs of the edges traversed, is equivalent to the objective value for the corresponding MVDRP solution. Therefore, the shortest path across G' from $v'_{0,0}$ to $v'_{0,|C|}$ is the smallest objective value possible in MVDRP while obeying *VisitOrder*. □

Corollary 1. *If we apply Phase 2 and Phase 3 of RTS to every possible permutation of VisitOrder, one of the solutions produced will be a globally optimal solution for MVDRP.*

We note that Theorem 1 and Corollary 1 do not apply to k-MVDRP in general, because the block partitioning scheme does not always construct an optimal assignment of drones to customers within operations.

4.1. Valid lower bounds

Solving the formulation from Section 2.4 proved to be intractable. Therefore, we attempted to construct various lower bound formulations, many of which were also intractable. However, we present two valid lower bound formulations that were tractable and relate k-MVDRP to existing problems in the literature.

4.1.1. Lower bound derived from a close-enough traveling salesman problem

Suppose that td is symmetric: $td(v_i, v_j) = td(v_j, v_i), \forall v_i, v_j \in V \cup C$. Also suppose that the energy function e only depends on the weight of packages carried. In any feasible k-MVDRP solution, for each customer $c_i \in C$, it is necessary that the truck must pass through at least one location $v_j \in V$ such that $td(v_j, c_i)/\alpha * e(w_i) + td(c_i, v_j)/\alpha * e(0) \leq EMAX$. Equivalently, the truck passes through a v_j such that $td(v_j, c_i) \leq \alpha * EMAX/(e(w_i) + e(0))$.

Let us set $cov(v_j, c_i) = 1$ whenever $td(v_j, c_i) * e(w_i) + td(c_i, v_j) * e(0) \leq EMAX$, and $cov(v_j, c_i) = 0$ otherwise. If $cov(v_j, c_i) = 1$, we say v_j covers c_i. We can form a lower bound on the k-MVDRP objective value by finding the least cost closed tour that starts at v_0, ends at v_0, and for each $c_i \in C$, goes through at least one v_j such that v_j covers c_i.

There must exist some location along the truck route such that the drone can fly from the truck to the customer location and return to the truck without running out of battery. The problem is similar to the close-enough traveling salesman problem (CETSP) (Wang et al., 2019), which requires a tour pass within a specified radius of each customer. In our case, any k-MVDRP tour must visit a vertex within radius $EMAX/(e(w_i) + e(0))$ of each customer $c_i \in C$. A formulation of the lower bound to the k-MVDRP is given below.

$$\texttt{Minimize:} \sum_{v_i \in V} \sum_{v_j \in V} tt(v_i, v_j) * x(v_i, v_j) \tag{C7}$$

`subject to:`

$$\sum_{v_i \in V} \sum_{v_k \in V} cov(v_i, c_j) * x(v_k, v_i) \geq 1, \forall j \in \{1, 2, \ldots, |C|\} \tag{C8}$$

$$\sum_{v_i \in V} x(v_i, v_j) = \sum_{v_k \in V} x(v_j, v_k), \forall v_j \in V \tag{C9}$$

$$\texttt{subtour elimination constraints} \tag{C10}$$

$$x(v_i, v_j) \in \{0, 1\}, \forall v_i, v_j \in V. \tag{C11}$$

Objective (C7) minimizes the length of the truck tour. Constraints (C8) ensure that the truck enters at least one $v_i \in V$, such that v_i serves as a cover for c_j. Constraints (C9) ensure conservation of flow. Constraints (C10) are generic subtour elimination constraints. Constraints (C11) ensure a binary selection of edges for the truck tour.

4.1.2. Lower bound derived from the min–max mTSP

Consider a relaxation of the k-MVDRP, where each drone has infinite battery capacity and range. Also, assume that a drone has no payload weight restrictions and that the flight time for a drone between any pair of locations is less than the driving of the truck between the pair of locations (i.e., $td(i, j) \leq tt(i, j)$, $\forall i, j \in V \cup C$). Thus, the drone has no incentive to ever return to the truck mid-route; it makes sense to proceed directly to the next customer due to the triangle inequality. An optimal solution to this relaxation would see the drone depart the truck immediately at the depot, and return to the truck at the depot only after it has completed all of its assigned deliveries. Therefore, this relaxation of the k-MVDRP is equivalent to the multiple traveling salesman problem (mTSP) where k is the number of vehicles deployed and the cost between pairs of locations are equal to the drone's flight time between the locations. So, under the condition $td(i, j) \leq tt(i, j)$, $\forall i, j \in V \cup C$, we can use a min-max mTSP formulation to compute a lower bound for the objective value of k-MVDRP.

5. Computational results

In Section 5.1, we discuss the instance generation procedures. In Section 5.2, we describe the sets of parameters on which we tested each instance. In Section 5.3, we present results. In Section 5.4, we analyze the results.

All computations were performed on a computer with an Intel i7-8750H processor operating at 2.20 GHz and containing 16GB of memory. Code was implemented in Python 2.7 and Gurobi 8.1.0 was used to solve for a TSP solution in Phase 1 of the RTS heuristic. Code used for the MVDRP can be found at http://stefan-poikonen.net/mvdrp/.

5.1. Instance generation procedures

For each combination of $|V| = 25, 50$ and $|C| = 25, 50$, we created 25 random instances. Thus, a total of 100 random instances were generated.

All launch/retrieval locations (V) and all customer locations (C) were chosen randomly from a uniform distribution over a 25 km by 25 km square region. The depot location was chosen randomly among V. The mass of packages demanded by customers is chosen from a uniform random distribution between 0 and 2.3 kg, which relates to statements made by Bezos (2013) about the size of packages that would possibly be subject to drone delivery. The truck is assumed to move at 5 m per second (or about 12 miles per hour). Both truck and drone travel times are computed as the Euclidean distance divided by the travel speed of the vehicle.

5.2. Tested drone parameters

5.2.1. Quadrocopter vs. octocopter

Stolaroff and Joshuah (2018) conduct an extensive study of energy use and greenhouse emissions in drones for commercial package delivery. We use their work as the basis for many of our model parameters. They describe two base cases: one for a smaller four rotor drone (quadrocopter) and one for a larger eight rotor drone (octocopter). The quadrocopter possesses a 1 kg battery; the octocopter possess a 10 kg battery.

In Supplemental Table 2 of Stolaroff and Joshuah (2018), a plot shows estimated energy use per meter traveled against the mass of packages carried for both quadocopter and octocopter cases based on their analytical model. We use piecewise linear functions, which are nearly identical to the energy estimates from Stolaroff and Joshuah (2018) and displayed in Fig. 4, to estimate the energy use of a drone carrying a given mass of packages between any pair of locations. We fix the constant HOV to be equal to the energy usage rate of the drone, while carrying no package. Exact equations for Fig. 4a and b can be found in Appendix A.

5.2.2. Drone speed

In both base cases of Stolaroff and Joshuah (2018), the drones move at a fixed speed of 10 m per second, which is slightly below the most energy efficient speed for both drones based on current designs. We test two drone speeds: 10 m per second and 15 m per second. The choice of 10 m per second is meant to simulate current capabilities. The higher 15 m per second speed is meant to simulate a hypothetical future drone that is more efficient (e.g., better power efficiency of engines, reduced drag). The power output does not change according to the speed choice of the drone.

5.2.3. Energy density of battery

In Stolaroff and Joshuah (2018), it is claimed that 540,000 Joules per kilogram approximates energy density that may be found in current drone batteries. They also estimate that with some foreseeable improvements of battery technology, the energy density may rise to 900,000 Joules per kilogram. We test both 540,000 J/kg and 900,000 J/kg.

5.2.4. Number of drones

For each instance and for each choice of parameters above, we computed routes for $k = 1, 2, \ldots, 8$ drones using the RTS heuristic algorithm.

5.3. Results

Among the 100 instances tested, eight of them were infeasible for at least some set of parameters. To allow comparison of results, the reported objective values and computational times are averages among the 92 of 100 instances (i.e., 25 instances each for $|V| = 25, 50$ and $|C| = 25, 50$) that were feasible for all tested parameters.

In Table 2, we present computational results. Each row represents results for a different set of input parameters. The column Drone Speed reports the speed of the drone in meters per second. The column Energy Density reports the energy density in Joules per kilogram. The column Rotors has value 4 for a quadrocopter and has value 8 for an octocopter. The column Obj. displays the average objective value among the 92 feasible instances found by applying the RTS algorithm. The objective value is measured in seconds of route duration. The column Time reports the average computational time of the RTS algorithm among the 92 feasible instances. The column Feas. reports the number of instances that were feasible among the set of 100 generated instances. The column LB reports the average value among feasible instances of the problem lower bound, which is computed as the maximum of the two lower bounds described in Sections 4.1.1 and 4.1.2. We used an exact formulation in Gurobi to solve for the lower bounds in Section 4.1.1. When computing the mTSP lower bound from Section 4.1.2, we used LKH3 (Helsgaun, 2017) to find heuristic mTSP solutions to reduce computation time for the lower bounds. These mTSP solutions are not guaranteed to be exact, however, LKH3 has been benchmarked on 127 mTSP instances, and in 124 cases found the best known solution. In all cases, lower bounds were formed in less than five minutes. In all benchmark instances of size 50 or smaller (i.e, the size of instances we use in Table 2), LKH3 found the best known solution. That is, we believe these heuristic mTSP solutions are likely to very closely approximate the exact mTSP solutions.

In Fig. 5, we plot the distributions of objective values among the 92 feasible instances. In blue are the average objective values found among computations where the drone speed was 10 m per

Table 2
Computational results.

Drone speed	Energy density	Rotors	k	Obj.	Time	Feas.	LB
10	540,000	4	1	26676.98	0.55	92	12512.10
10	540,000	4	2	19498.40	0.61	92	8788.207
10	540,000	4	3	16922.58	0.64	92	7971.14
10	540,000	4	4	15504.69	0.68	92	7655.66
10	540,000	4	5	14732.36	0.71	92	7512.26
10	540,000	4	6	14272.36	0.75	92	7462.53
10	540,000	4	7	14019.66	0.78	92	7441.64
10	540,000	4	8	13829.10	0.80	92	7435.88
10	540,000	8	1	24991.71	0.70	100	12434.81
10	540,000	8	2	18089.47	0.95	100	7184.09
10	540,000	8	3	15433.59	1.14	100	5774.88
10	540,000	8	4	13744.08	1.36	100	5262.12
10	540,000	8	5	12603.37	1.57	100	5041.90
10	540,000	8	6	11823.73	1.80	100	4968.10
10	540,000	8	7	11195.88	2.01	100	4937.39
10	540,000	8	8	10678.31	2.20	100	4928.92
10	900,000	4	1	25088.91	0.72	100	12434.81
10	900,000	4	2	18100.74	1.03	100	7187.70
10	900,000	4	3	15406.86	1.27	100	5828.64
10	900,000	4	4	13713.32	1.55	100	5352.04
10	900,000	4	5	12552.28	1.81	100	5147.24
10	900,000	4	6	11749.28	2.10	100	5078.81
10	900,000	4	7	11095.76	2.36	100	5050.78
10	900,000	4	8	10508.18	2.66	100	5043.10
10	900,000	8	1	24830.67	1.37	100	12434.81
10	900,000	8	2	17820.03	2.54	100	7181.51
10	900,000	8	3	15076.49	3.64	100	5709.28
10	900,000	8	4	13325.06	4.84	100	5140.18
10	900,000	8	5	12134.37	6.00	100	4881.49
10	900,000	8	6	11237.77	7.30	100	4791.79
10	900,000	8	7	10513.51	8.49	100	4754.11
10	900,000	8	8	9939.76	9.79	100	4743.73
15	540,000	4	1	20801.54	0.57	100	8330.60
15	540,000	4	2	14993.71	0.69	100	5454.47
15	540,000	4	3	12791.37	0.78	100	4872.59
15	540,000	4	4	11426.95	0.89	100	4658.15
15	540,000	4	5	10538.31	0.99	100	4560.67
15	540,000	4	6	9921.43	1.09	100	4526.87
15	540,000	4	7	9467.92	1.19	100	4512.67
15	540,000	4	8	9152.89	1.29	100	4508.76
15	540,000	8	1	19477.77	0.92	100	8289.87
15	540,000	8	2	13864.34	1.49	100	4787.67
15	540,000	8	3	11550.42	2.01	100	3806.19
15	540,000	8	4	10149.67	2.57	100	3426.79
15	540,000	8	5	9226.61	3.11	100	3254.86
15	540,000	8	6	8534.77	3.72	100	3195.71
15	540,000	8	7	7947.31	4.26	100	3170.86
15	540,000	8	8	7477.61	4.84	100	3164.01
15	900,000	4	1	19790.31	0.94	100	8289.87
15	900,000	4	2	14000.15	1.58	100	4787.67
15	900,000	4	3	11626.60	2.19	100	3806.19
15	900,000	4	4	10189.43	2.85	100	3426.79
15	900,000	4	5	9229.91	3.50	100	3255.34
15	900,000	4	6	8502.86	4.24	100	3196.19
15	900,000	4	7	7934.69	4.90	100	3171.35
15	900,000	4	8	7464.02	5.62	100	3164.50
15	900,000	8	1	19314.56	2.16	100	8289.87
15	900,000	8	2	13609.35	4.55	100	4787.67
15	900,000	8	3	11170.28	7.05	100	3806.19
15	900,000	8	4	9683.30	9.63	100	3426.79
15	900,000	8	5	8611.47	12.29	100	3254.33
15	900,000	8	6	7901.87	15.18	100	3194.53
15	900,000	8	7	7280.22	17.83	100	3169.41
15	900,000	8	8	6830.86	20.51	100	3128.11

second. In orange are the average objective values found among computations where the drone speed was 15 m per second.

In Fig. 6, we plot the distributions of objective values among the 92 feasible instances. In blue are the average objective values found among computations where the battery energy density was 540,000 Joules per kilogram. In orange are the average objective values found among computations where the battery energy density was 900,000 Joules per kilogram.

In Fig. 7, we plot the distributions of objective values among the 92 feasible instances. In blue are the average objective values found using quadrocopters. In orange are the average objective values found using octocopters.

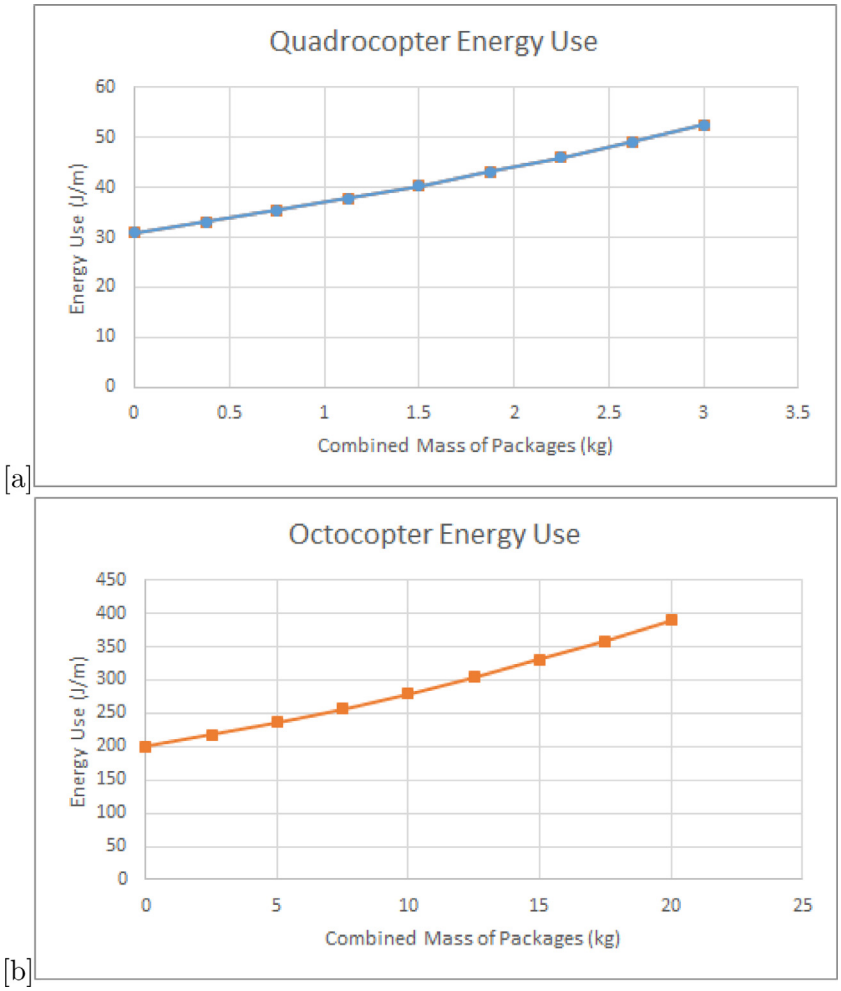

Fig. 4. Energy dissipation in Joules per meter traveled (vertical axis) is plotted against the combined mass of packages carried by the drone (horizontal axis), assuming 10 m per second flight speed. In (a), energy dissipation rates are shown for a quadrocopter with a 1 kg battery. In (b), energy dissipation rates are shown for an octocopter with a 10 kg battery.

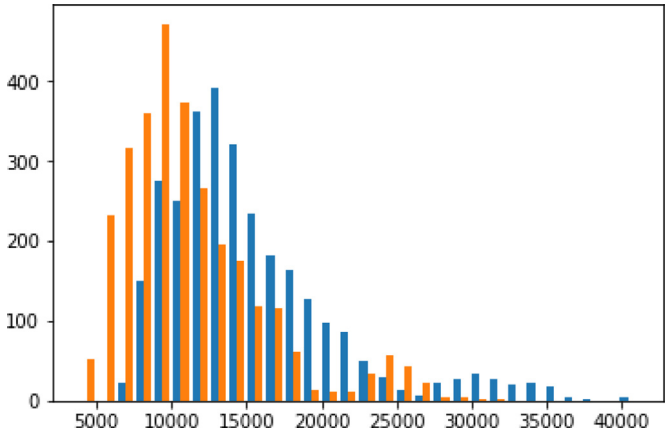

Fig. 5. A comparison of objective values based on drone speed: 10 m per second (blue) vs. 15 m per second (orange). The average objectives values were 15222.16 and 11264.45, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

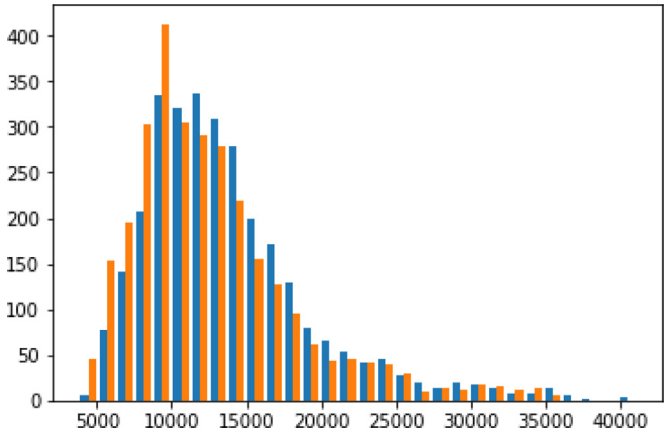

Fig. 6. A comparison of objective values based on battery energy density: 540,000 J/kg (blue) vs. 900,000 J/kg (orange). The average objectives values were 13791.84 and 12694.78, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In Fig. 8, we plot the distributions of objective values among the 92 feasible instances based on the number of drones.

5.4. Analysis of results

In general, the computational time required to run the RTS algorithm was very small, averaging only 3.26 s in total. On the largest instances ($|V| = |C| = 50$), the average computational time was only 7.32 s.

For a fixed number of drones, objective values showed the greatest sensitivity to drone speed. With drone speed at 15 m per second, the average objective value was 26.0% lower compared to a drone speed of 10 m per second.

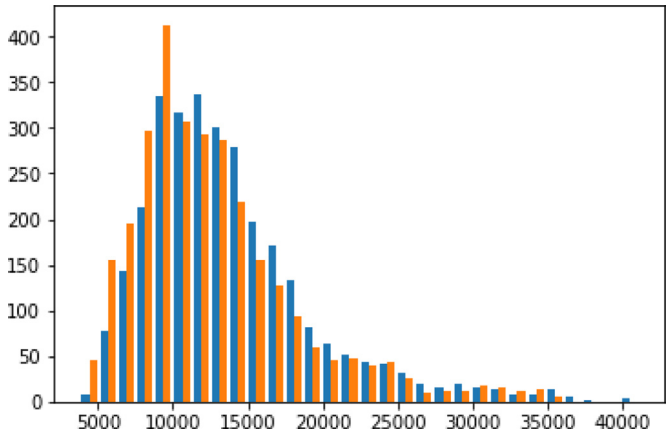

Fig. 7. A comparison of objective values based drone type: quadrocopter (blue) vs. octocopter (orange). The average objectives values were 13796.99 and 12689.63, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

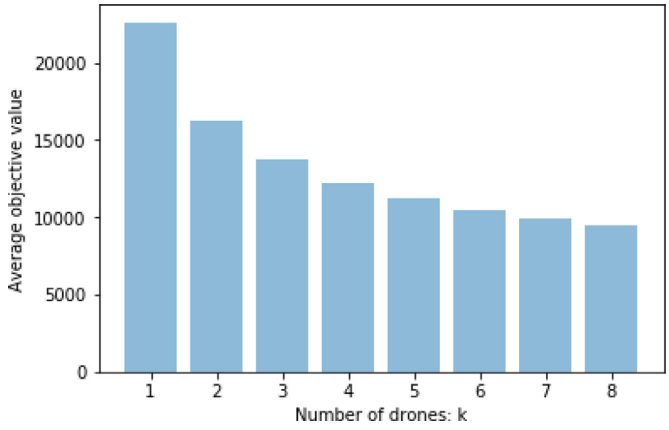

Fig. 8. A comparison of objective values by number of drones, ranging from $k = 1$ to 8.

Improved battery technology and using larger octocopters, which have a larger effective range, also improved results by 8.0% and 8.0%, respectively.

The number of drones on the truck was also very important. Using $k = 8$ drones, the maximum tested, resulted in objective value decreases of 58.1% relative to a single drone. However, decreasing marginal returns was certainly observed.

Averaged across all instances, the gap between the objective value and lower bound was quite large, averaging 56.0%. The two lower bounds used each have an inherent weakness. The CETSP-derived lower bound does not account for any time the truck spends waiting for one or more drones. On the other hand, the mTSP-derived lower bound does not account for limited payload capacity and battery restrictions of the drone. Of the two lower bounds, the mTSP-derived lower bound was larger in 85.8% of instances, whereas the CETSP-derived lower bound was larger in 14.2% of instances. As the number of drones increased, the CETSP-derived lower bound was larger than the mTSP-derived lower bound at a slightly higher rate. When $k = 1$, the CETSP-derived lower bound was the better lower bound in only 1.9% of the instances. When $k = 8$, the CETSP-derived lower bound was the better lower bound in 17.9% of the instances. Other attempted formulations that integrated both synchronization issues (e.g., waiting times) and energy constraints proved to be computationally intractable.

For each value of $k = 1, 2, \ldots, 8$, we conducted three separate Wilcoxon Signed-Rank Tests which compared median objective values for instances using (1) quadrocopter vs. octocopter, (2) current battery technology vs. future battery technology, and (3) drone speed of 10 m/s vs. 15 m/s. In all cases, the resulting p-value was less than 0.01, strongly suggesting a significant difference in medians in each case.

6. Conclusions and future work

We introduced the k-MVDRP. The underlying model is novel because it relaxes a number of simplifying constraints that often appear in the literature. In particular, we allow a drone to carry multiple heterogeneous packages, we allow the energy drain function to be any non-decreasing function of weight for each location pair, we decouple the set of launch/landing vertices from the set of customer locations, and we allow an arbitrary number of drones on a truck.

Computational experiments were conducted using several physical parameters that reflect recent research about drone energy efficiencies, velocity, and payload capacity. The results indicate that objective values are highly sensitive to drone speed. The number of drones was also very impactful on the objective value, though decreasing marginal savings were observed.

We believe there are several potential directions for future research. Our heuristic algorithm may introduce suboptimality in two ways. Firstly, the optimal customer visitation order may not be the same as in the Euclidean TSP, and secondly, the assignment of drones to customers within an operation is chosen suboptimally by the block partitioning scheme. We believe exploration of other potential customer visitation orders and using a different assignment mechanism within an operation (perhaps using an integer programming formulation) are two directions that may improve solution quality. Other completely different algorithmic approaches may prove fruitful also. Additionally, we would like to find tractable formulations that strongly enhance the lower bounds and better account for the structure of this problem, inclusive of both range and synchronicity constraints.

We would also like to consider a number of variants. In one such variant, the speed of the drone is variable. In particular, we would like to use the RTS heuristic framework, but allow the speed of each drone on an operation to be a variable that may be optimized. We could find the maximum speed of the drone on the operation such that it does not run out of battery before rendezvousing with the truck. Another variant would allow the truck to make deliveries while one or more drones is airborne.

Appendix A

The energy function for the quadrocopter is $e(v_i, v_j, W) =$

$$\begin{cases} 31.000 + (33.110 - 31.000)(W - 0.000)/0.375 & 0.000 \le W \le 0.375 \\ 33.110 + (35.364 - 33.110)(W - 0.375)/0.375 & 0.375 \le W \le 0.750 \\ 35.364 + (37.712 - 35.364)(W - 0.750)/0.375 & 0.750 \le W \le 1.125 \\ 37.712 + (40.342 - 37.712)(W - 1.125)/0.375 & 1.125 \le W \le 1.500 \\ 40.342 + (43.088 - 40.342)(W - 1.500)/0.375 & 1.500 \le W \le 1.875 \\ 43.088 + (46.021 - 43.088)(W - 1.875)/0.375 & 1.875 \le W \le 2.250 \\ 46.021 + (49.154 - 46.021)(W - 2.250)/0.375 & 2.250 \le W \le 2.625 \\ 49.154 + (52.500 - 49.154)(W - 2.625) & 2.625 \le W \le 3.000 \end{cases}$$

and is measured in Joules per meter.

The energy function for the octocopter is $e(v_i, v_j, W) =$

$$\begin{cases} 200.00 + (217.41 - 200.00)(W - 0.000)/2.5 & 0.000 \le W \le 2.5 \\ 217.41 + (236.34 - 217.41)(W - 2.5)/2.5 & 2.5 \le W \le 5.0 \\ 236.34 + (256.92 - 236.34)(W - 5.0)/2.5 & 5.0 \le W \le 7.5 \\ 256.92 + (279.28 - 256.92)(W - 7.5)/2.5 & 7.5 \le W \le 10.0 \\ 279.28 + (303.60 - 279.28)(W - 10.0)/2.5 & 10.0 \le W \le 12.5 \\ 303.60 + (330.03 - 303.60)(W - 12.5)/2.5 & 12.5 \le W \le 15.0 \\ 330.03 + (358.77 - 330.03)(W - 15)/2.5 & 15.0 \le W \le 17.5 \\ 358.77 + (390.00 - 358.77)(W - 17.5) & 17.5 \le W \le 20.0 \end{cases}$$

and is measured in Joules per meter.

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 52.4, 965–981.

Air, A. P., 2018. Amazon prime air: frequently asked questions. (Accessed 9 April 2018), https://www.amazon.com/b?node=8037720011.

Bezos, J., 2013. Interview with charlie rose. Broadcasted on: 60 Minutes, CBS. December 1, Transcript: https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/.

Boeing YouTube Channel, January 2018. Future of Autonomous Air Travel: Boeing Unveils New Cargo Air Vehicle Prototype. Boeing. (Accessed 29 April 2018).

Borak, M., January 2018. World's top donre seller DJI made $2.7 billion in 2017. TechNode. January 3 (Accessed 7 April 2018). https://technode.com/2018/01/03/worlds-top-drone-seller-dji-made-2-7-billion-2017/.

Campbell, J., Sweeney, D., Zhang, J., 2017. Strategic Design for Delivery with Trucks and Drones. Technical Report.

Carlsson, J., Song, S., 2017. Coordinated logistics with a truck and a drone. Manag. Sci. 64.9, 4052–4069.

Cary, N., Bose, N., September 2016. UPS, Fedex and Amazon Gather Flight Data to Prove Drone Safety. Venture Beat. (Accessed 17 May 2017).

Davies, A., January 2018. Boeing's Experimental Cargo Drone is a Heavy Lifter. Wired.com. (Accessed 4 September 2018).

Dayarian, I., Savelsbergh, M., Clarke, J.P., 2018. Same-day Delivery With Drone Resupply. Technical Report. Available at: http://www.optimization-online.org/DB_FILE/2017/09/6206.pdf.

DJI, 2018a. Chart: Matrice 600 pro Flight Time vs. Payload (kg). (Image). Accessed 4 September 2018)

DJI, 2018b. Spreading wings s900. (Accessed 4 September 2018), https://www.dji.com/spreading-wings-s900.

Flynt, J. How much weight can a drone carry? November 2017. (Accessed 1 August 2018), https://3dinsider.com/drone-payload/3DInsider.com.

Franco, C.D., Buttazzo, G.C., 2015. Energy-aware coverage path planning of UAVs. ICARSC. Available at: https://www.researchgate.net/profile/Carmelo_Di_Franco/publication/279848193_Energy-aware_Coverage_Path_Planning_of_UAVs/links/57d7c2e908ae6399a3973197/Energy-aware-Coverage-Path-Planning-of-UAVs.pdf.

Ha, Q., Deville, Y., Pham, Q., Há, M., 2018. On the min-cost traveling salesman problem with drone. Transp. Res. Part C: Emerg.Technol. 86, 597–621.

Helsgaun, K., 2017. An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems. Technical Report. Roskilde, Denmark

Lillian, B., May 2017. Flying drones beyond line of sight: how can we get there? Unmanned aerial. (Accessed 22 March 2018), https://unmanned-aerial.com/flying-drones-beyond-line-sight-can-get.

Murray, C., Chu, A., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp. Res. Part C: Emerg.Technol. 54, 86–109.

Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: extended models and connections. Networks 70.1, 34–43.

Poikonen, S., Wasil, E., Golden, B., 2019. A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS J. Comput. 31.2, 335–346.

Reuters staff, September 2015. Finnish Post Office Tests Drone for Parcel Delivery. Reuters. (Accessed 11 June 2018).

Reuters staff, April 2018. Russian Postal Drone Program Hits Wall in Debut. Reuters. (Accessed 16 April 2018).

Rey, P., 2016. "paketzustellung per drohne: DPDgroup startet den weltweit ersten drohnenverkehr im linienbetrieb" (package delivery with a drone: DPDgroup starts the first regular drone service worldwide). Press Release DPDgroup. https://www.dpd.com/portal_de/home/news/aktuelle_neuigkeiten/dpdgroup_startet_den_weltweit_ersten_drohnenverkehr_im_linienbetrieb.

Spangler, T., 2018. Amazon has More Than 100 Million Prime Subscribers, Jeff Bezos Discloses. Variety.

Stolaroff, Joshuah, K., et al., 2018. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. Nat. Commun. 9.1, 409.

Ulmer, M., Thomas, B., 2018. Same-day delivery with a heterogeneous fleet of drones and vehicles. Networks 72.4, 475–505.

UPS YouTube Channel, February 2017. UPS tests residential delivery via drone. (Accessed March 2017) https://www.youtube.com/watch?v=xx9_6OyjJrQ. Note: YouTube Video.

Wang, X., Golden, B., Wasil, E., 2019. A steiner zone variable neighborhood search heuristic for the close-enough traveling salesman problem. Comput. Oper. Res. 101, 200–219.

Wang, X., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: several worst-case results. Optim. Lett. 11.4, 679–697.

Wing, P., 2018. Project wing. (Accessed 15 June 2018) https://x.company/projects/wing/.