Fleet Design Optimization of Package Delivery Unmanned Aerial Vehicles Considering Operations

Shugo Kaneko[1] and Joaquim R. R. A. Martins[2]
University of Michigan, Ann Arbor, MI, 48109

Abstract

The conceptual design process of aircraft starts by deciding the representative mission requirements, followed by optimization of design variables to satisfy the given requirements. However, the appropriate mission requirements are not obvious, especially when designing package delivery UAVs (also called *drones*). The UAVs must accommodate various combinations of package weights and delivery distances. The complexity increases further when designing a heterogeneous fleet of UAVs that serves a large number of customers. This work addresses this problem by solving coupled design-operation optimization to find optimal mission requirements and optimal UAV designs simultaneously. We formulate this problem as mixed-integer nonlinear optimization and propose a sequential heuristic algorithm to solve the coupled problem. The benchmark study of the proposed algorithm against a nonconvex branch-and-cut solver shows that the sequential heuristics are effective. We also demonstrate that the simultaneous UAV design and routing optimization reduces the UAV weight across the fleet by more than 12% on average compared to the conventional baseline.

Nomenclature

UAV Conceptual Design Parameters

A	=	rotor disk area, m^2
AR	=	wing aspect ratio
C_D	=	drag coefficient
C_L	=	lift coefficient
C_T	=	thrust coefficient
D	=	drag, N
E_{req}	=	energy required, J
e	=	Oswald efficiency
$f_{\text{sizing}}(\cdot)$	=	sizing model as a function mapping
J	=	propeller advance ratio
n_c	=	number of customers
n_{rev}	=	rotor revolution, 1/s
n_{rotor}	=	number of rotors
P	=	power, W

[1] Ph.D. Candidate, Department of Aerospace Engineering. Student Member AIAA.

[2] Pauline M. Sherman Collegiate Professor, Department of Aerospace Engineering. Fellow AIAA.

P_0	=	rotor profile power, W
R	=	flight range, m
r	=	rotor radius, m
S_b	=	body reference area, m^2
S_w	=	wing area, m^2
T	=	thrust, N
t	=	duration, s
V_∞	=	cruise speed, m/s
V_i	=	rotor induced velocity, m/s
W	=	weight, N
β	=	rotor shaft tilt angle, rad
η_{hover}	=	hover figure of merit
κ	=	induced power factor
λ	=	rotor inflow ratio
μ	=	rotor edgewise advance ratio
ρ	=	air density, kg/m^3
ρ_b	=	battery energy density, Wh/kg
σ	=	rotor solidity
Ω	=	rotor angular velocity, rad/s

Vehicle Routing Parameters and Sets

d_{ij}	=	distance between node i and j, m
M	=	large constant
n_{design}	=	number of different UAV designs in the fleet
n_{UAV}	=	maximum number of UAVs available in the fleet
p	=	number of the UAVs of each design in the fleet
q_i	=	demand by customer i, kg
x_{ijk}	=	binary variable relating arcs and vehicles
y_{ik}	=	binary variable relating nodes and vehicles
α_k	=	binary variable indicating vehicle activity
$\mathcal{N}$	=	set of customers
$\mathcal{S}$	=	subset of customers
$\mathcal{V}$	=	set of customers and depot
$\mathcal{A}$	=	set of arcs between nodes
$\mathcal{K}$	=	set of all vehicles
$\mathcal{K}^{\text{Hexa}}$	=	set of hexarotor UAVs
$\mathcal{K}^{\text{QBiT}}$	=	set of QBiT UAVs

Vehicle Routing Indices, Subscripts, and Superscripts

i, j	=	node index, $i, j \in \mathcal{V}$
k	=	vehicle index, $k \in \mathcal{K}$
o, d	=	depots
pc	=	power in cruise
ph	=	power in hover
w	=	weight
H	=	hexarotor
Q	=	QBiT
UB	=	upper bound

Other Parameters

f^*	=	optimized objective value
f_0	=	baseline objective value
X	=	UAV design vector in payload-range space

1 Introduction

Unmanned aerial vehicles (UAVs) are viable for transporting commercial packages and medical supplies. The advantages of UAVs—also known as *drones*—over conventional ground vehicles include rapid last-mile delivery and potentially lower impact on the environment [1]. In particular, electric vertical takeoff and landing (eVTOL) UAVs are a potential means for lightweight package delivery. The electric propulsion is essential to build lightweight UAVs, and the hovering and vertical flight capability enable direct delivery to customers' backyards or rooftops.

One of the active research fields on the eVTOL UAV is the conceptual design methodology, which is not yet established and not readily available in the open literature because of the lack of statistical data on existing vehicles as opposed to fixed-wing aircraft. For package delivery applications, Sridharan et al. [2] performed a conceptual sizing study of quadrotor biplane tail-sitter (QBiT) configuration [3] for various payload weights and mission ranges. Govindarajan and Sridharan [4] presented an optimization-based conceptual design approach and applied it to four eVTOL configurations, including QBiT and hexarotor. Several researchers also investigated multirotor designs with no wing [5–7].

Regardless of the model or aircraft configuration, the conceptual design process starts by setting appropriate mission requirements, for which engineers optimize the sizing variables. The research question here is the following: How should engineers choose appropriate mission requirements? This is a challenging task for a package delivery UAV because the UAV needs to perform delivery missions of various package weights and delivery distances. The problem becomes even more complicated when we design a fleet of UAVs that serves a large number of customers. In this work, we address this challenge using multidisciplinary design optimization (MDO) to optimize the mission and UAV designs simultaneously. We achieve this goal by optimizing the UAV fleet design and delivery operations concurrently to reduce fleet acquisition and operating costs.

The operational optimization of package delivery is known as the vehicle routing problem (VRP), which is an integer optimization problem that seeks the optimal routes of vehicles. The VRP has traditionally been studied for truck delivery. Recently, there have also been research efforts on UAV routing [8–10]. The recent literature includes truck-UAV collaborative delivery [11–13] and the UAV-only delivery [14–17]. One crucial aspect of UAV routing problems is energy consumption modeling. Because of the limited energy capacity of UAVs, the range or endurance is a limiting factor for UAVs, unlike conventional trucks. Dorling et al. [14] solved a UAV routing problem using a linearization of an energy consumption model based on the momentum theory in hover. They approximated the hovering power to be linear with respect to the vehicle weight. Cheng et al. [17] extended their work by incorporating a nonlinear energy model into UAV routing. Coelho et al. [15] assumed that the UAV power consumption is linear with respect to the flight speed and constant with respect to the weight in their routing model. Other authors did not explicitly compute the energy consumption to simplify the routing problem formulation; instead, they replaced the energy capacity constraint with a flight range constraint [11, 12] or endurance constraint [13, 16]. Zhang et al. [18] summarized the energy consumption models used in the recent UAV literature. The energy models also serve as a means to evaluate the environmental aspect of UAV delivery. Chiang et al. [12] assessed the sustainability impact of delivery by solving the UAV routing problem. Stolaroff et al. [1] and Kirschstein [19] compared UAV delivery with ground delivery, but these studies did not include UAV routing.

The literature mentioned above solved VRPs to make short-term tactical decisions, i.e., to decide on an upcoming delivery schedule using an existing fleet. Instead of tactical decisions, our goal in this paper is to make a long-term strategic decision on the UAV fleet design via VRP. This type of VRP for a strategic decision is categorized as *fleet size and mix VRP* (FSMVRP) [20, 21], which aims to find the optimal number and composition of vehicles in the fleet to maximize long-term profit. Accordingly, the customer information on which an FSMVRP is solved should represent the demand over the fleet life cycle. Compared to the rich literature on ordinary VRPs for tactical routing, the FSMVRP has drawn less attention from the research community. There has been a limited number of UAV FSMVRP literature [22, 23] that only performed homogeneous fleet sizing. In the current work, we expand the scope of FSMVRP by coupling it to the UAV conceptual design optimization. To the authors' best knowledge, there has been no literature on the truck or ship FSMVRP that includes the vehicle or vessel design variables. This is not a surprise, given that delivery or transportation companies do not usually have the freedom to design vehicles or vessels by themselves. However, because of the low unit cost of UAVs, the delivery service providers may decide to design the

UAVs for their own use. The only previous work addressing the UAV design-routing optimization is by Choi [23]. They performed sequential optimization of multi-trip VRP and UAV design on a 20-node problem with up to 3 homogeneous UAVs. In the current work, we aim to expand the problem scope by considering a heterogeneous fleet, including more UAV design variables (such as cruise speed and payload capacity), and solving larger-scale problems.

Although the application is not package delivery, there have been several efforts on simultaneous optimization of commercial aircraft design and operations. Taylor and de Weck [24] used simulated annealing in combination with a linear programming solver to optimize the aircraft conceptual design and cargo network flow. Mane et al. [25] proposed a sequential decomposition approach to solve the aircraft sizing and airline allocation optimization, and they compared their approach to a mixed-integer nonlinear programming (MINLP) exact solver. Davendralingam and Crossley [26] used the decomposition method by Mane et al. [25] to perform robust optimization of aircraft design and airline network design. Jansen and Perez [27] also optimized the aircraft design and fleet allocation by decomposing the design-allocation coupled problem. In contrast to the decomposition-based approaches in the literature cited above, Hwang et al. [28] used a monolithic MDO architecture [29] to solve the coupled optimization of computational-fluid-dynamics-based aircraft design, airline allocation, and flight trajectory. They used continuous relaxation of the allocation problem to enable large-scale optimization. Roy et al. [30] proposed a surrogate-based monolithic optimization framework to solve mixed-integer design-allocation problems.

Our work shares the same overarching goal as the literature [23–26, 28, 30]: We concurrently optimize the vehicle design and fleet operation to achieve better system design. Our new contribution is that we solve VRP as an operation problem instead of solving resource allocation problems. The VRP, which is essential for UAV delivery, is more complicated than the resource allocation problem because the VRP decision variables include route selection from a network that often has an overwhelming number of possible combinations. In contrast, resource allocation problems do not typically include routing variables.

The outline of the paper is as follows. Section 2 provides the problem statement of the design-operation optimization for package delivery UAVs. Section 3 summarizes the eVTOL conceptual design model used in this study. In Section 4, we present an MINLP formulation for the design-operation optimization problem, and we propose an effective sequential heuristic algorithm (SHA). We then perform a benchmark study of the optimization approaches in Section 5. Section 6 compares the baseline and optimized solution to demonstrate the benefit of the coupled optimization. Section 6 also presents an optimization result for a delivery case with battery recharging.

2 Problem Description

2.1 Simultaneous Optimization of UAV Design and Operations

We propose to minimize the summation of the UAV takeoff weight across the fleet that serves a given set of customers by varying fleet design and operation variables. Minimizing the fleet weight reduces both the fleet acquisition cost and operating cost because (1) smaller vehicles likely cost less to manufacture, and (2) lighter vehicles likely consume less energy. Further discussion on the objective function selection is provided in Section 3.5. The operational optimization variables determine the allocation and routing; more precisely, the allocation of each UAV to a subset of customers and the sequence of the visit within the subset. The fleet design variables consist of the number of UAVs in the fleet and the design of each UAV. This work considers a heterogeneous fleet composed of multiple designs and two configurations, which are described in Section 3.

The design-operation optimization is a mixed-integer optimization problem because it has both discrete routing and continuous design variables. Furthermore, the UAV design optimization is generally nonlinear, making the coupled optimization an MINLP problem [31, Ch. 8].

Figure 1 shows the coupling structure of the design-operation optimization. The UAV fleet determines the optimal operations because the routing solution is subject to each UAV's payload and energy capacity constraints. The operational optimization outputs a set of routes, which gives the flight mission requirements for UAV design optimization. In this work, we parametrize the missions in terms of the delivery distance, total package weight, and the number of customers on each route. The UAV design optimization outputs a set of UAV designs, which compose a fleet, given the mission requirements. In summary, there is a cycle of

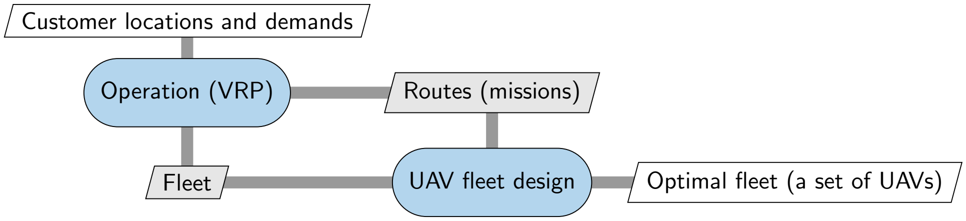


Figure 1: Coupling of UAV fleet design and operations.

information between the operation and design optimization; thus, we need coupled optimization to find the optimal fleet design.

2.2 Vehicle Routing Problem

This work solves an FSMVRP [20, 21] as an operational optimization problem for package delivery. The goal of FSMVRP is to make a strategic decision on the size and composition of a yet-to-be-deployed fleet. The FSMVRP solver determines the optimal UAV selection out of various UAV candidates for anticipated delivery services. Because of its long-term objective, FSMVRP requires representative customer information over the life cycle of the fleet, which we assume to be available. Specifically, the customer information consists of the customer locations and the amount (weight) of each customer's demand.

The key assumptions and constraints in our VRP model are as follows:

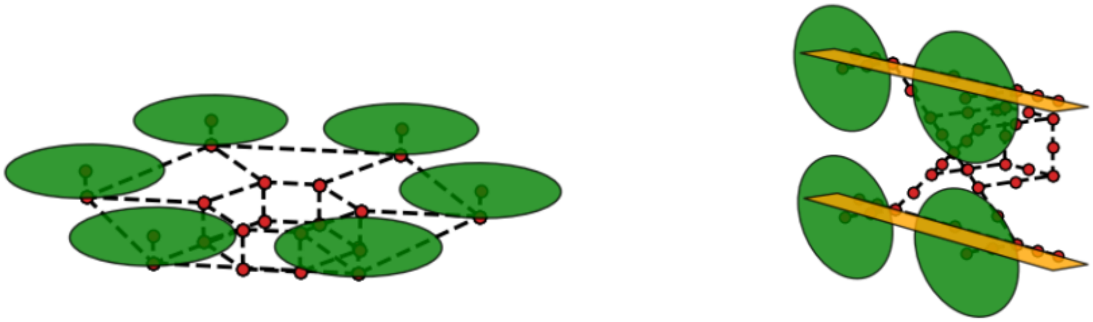
1. The fleet is heterogeneous.
2. Not all UAVs in the fleet need to be used unlike conventional VRP; the FSMVRP solver finds the optimal number and composition of UAVs.
3. Each UAV must satisfy the payload capacity and energy consumption constraints.
4. Customer locations and demands are deterministic and known a priori.
5. All customers must be served by exactly one UAV. Customers cannot be dropped or served more than once (i.e., we prohibit splitting the demand between multiple vehicles).
6. We only consider a single depot. Each route must be a closed loop that starts and ends at the depot.
7. Each UAV is used only once; multi-trip of a vehicle is not allowed. For the problem instance in Section 6.4, we allow UAVs to recharge batteries at recharging stations.

Assumptions 1–3 are essential for designing an effective UAV fleet, whereas 4–7 are adopted to simplify the routing problem.

3 UAV Conceptual Design Model

This section presents an eVTOL UAV sizing model used in this study. We consider two different configurations, a hexarotor and a QBiT [2], as shown in Fig. 2. Both configurations are capable of vertical climb, descent, and hover. For cruise, the hexarotor performs the edgewise flight, whereas the QBiT uses wings to generate the lift. The sizing models and parameters in this study were mainly based on the work of Govindarajan and Sridharan [4].

The sizing model aims to find the minimum-weight UAV design given mission requirements. We parametrize the mission in terms of range R, total payload weight W_{payload}, and the number of customers n_{c} on the route. Figure 3 shows an example of the mission profile. We optimize a UAV for multiple missions, i.e., the optimizer finds a single UAV design that can fly multiple delivery scenarios. In such cases, the mission inputs R, W_{payload}, and n_{c} are vectors. The sizing outputs are the UAV takeoff weight W_{total}, battery weight W_{battery}, and power consumption in hover P_{hover} and cruise P_{cruise}.



(a) Hexarotor

(b) Quadrotor biplane tail-sitter (QBiT)

Figure 2: Delivery UAV configurations considered in this study [4].[1]

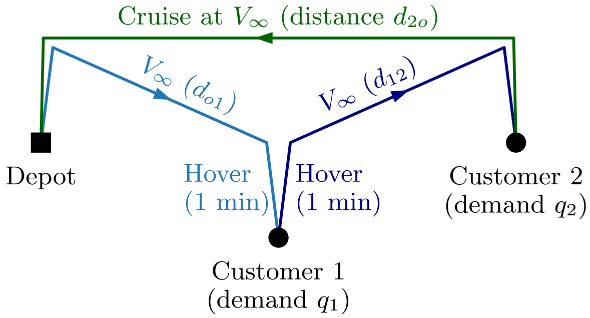

Cruise at V_∞ (distance d_{2o})
V_∞ (d_{o1})
V_∞ (d_{12})
Depot
Hover (1 min)
Hover (1 min)
Customer 2 (demand q_2)
Customer 1 (demand q_1)

Figure 3: Example of the mission profile for a 2-customer delivery, where $n_\mathrm{c} = 2$, the range is $R = d_{o1} + d_{12} + d_{2o}$, and the payload weight is $W_\mathrm{payload} = q_1 + q_2$.

The inputs and outputs of the sizing module are summarized as follows:

$$(W_\mathrm{total}, W_\mathrm{battery}, P_\mathrm{hover}, P_\mathrm{cruise}) = f_\mathrm{sizing}\left(R, W_\mathrm{payload}, n_\mathrm{c}\right) \ . \tag{1}$$

Inside the above function $f_\mathrm{sizing}(\cdot)$, we perform the UAV design optimization.

Figure 4 shows the extended design structure matrix (XDSM) [32] of the QBiT sizing model. The hexarotor sizing model also has a similar structure.

3.1 Weight Estimation

The UAV total weight consists of the payload, battery, and empty weight as follows:

$$W_\mathrm{total} = W_\mathrm{payload} + W_\mathrm{battery} + W_\mathrm{empty} \ . \tag{2}$$

The empty weight is further broken down into motor, electronic speed controller (ESC), rotor, and frame weight. The weights of each component are given by the following regression models:

$$\begin{aligned}
W_\mathrm{empty} &= W_\mathrm{motor} + W_\mathrm{ESC} + W_\mathrm{rotor} + W_\mathrm{wing} + W_\mathrm{frame} \ , && (3)\\
W_\mathrm{motor} &= (2.506 \times 10^{-4}) P \ , && (4)\\
W_\mathrm{ESC} &= (3.594 \times 10^{-4}) P \ , && (5)\\
W_\mathrm{rotor} &= 0.7484 r^2 - 0.0403 r \ , && (6)\\
W_\mathrm{wing} &= -0.0802 + 2.2854 S_w \ , && (7)\\
W_\mathrm{frame} &= 0.5 + \beta W_\mathrm{total} \ . && (8)
\end{aligned}$$

The motor and ESC weights are estimated using Eqs. (4) and (5) [4], where P is the installed power of the motor. We assume a 50% power margin on top of the maximum power required during the missions. The rotor weight is given by Eq. (6) [5], where r is the rotor radius. For the wing of the QBiT configuration, we assume the wing weight to be linear with respect to the wing area as Eq. (7). The frame weight in Eq. (8) consists of the structural weight and various weights such as wires and recovery parachutes. We also assume that the frame weight is linear to the takeoff weight, and we use $\beta = 0.20$ for the hexarotor and $\beta = 0.18$ for the QBiT. We determined the linear coefficients in Eqs. (7) and (8) using a least-squares linear regression to fit the weight estimation results by Govindarajan and Sridharan [4], where they used finite element analysis and sizing optimization to estimate the component weights. The smaller value of β for the QBiT accounts for the wings for which we compute the weight separately, which also carry the structural load. Eqs. (4)–(8) use SI units, i.e., Newtons for the weight, Watts for the power, and meters for length.

The battery weight is computed based on the battery energy density and the total energy required for the mission. We assume a simple mission profile composed of vertical takeoff, cruise, and landing. Furthermore, we approximate the power in vertical climb and descent to be the same as the power in hover on average, as suggested by Dorling et al. [14]. The energy required for the mission is then

$$E_{\text{req}} = P_{\text{hover}}\Big(2(n_c+1)t_{\text{hover}}\Big) + P_{\text{cruise}}\frac{R}{V_\infty}\ , \tag{9}$$

where V_∞ is the cruise speed; P_{hover} and P_{cruise} are the power consumption in hover and cruise, respectively. The first term is the energy for takeoff and landing, and the second is the energy for cruise. We assume the hovering time t_{hover} of 60 s for a takeoff or landing operation; therefore, a UAV hovers for 120 s at each customer or the depot. We also assume that P_{hover} and P_{cruise} are constant across the mission, which means that we ignore the UAV weight change after unloading the payload. This implies an extra margin on the UAV energy capacity. This simplification was necessary to reduce the routing model complexity and keep the problem solvable using an MINLP solver and routing heuristics. Once the energy is computed, the battery weight is given by $W_{\text{battery}} = E_{\text{req}}/0.85\rho_b$. The battery density ρ_b is assumed to be 158 Wh/kg, which is conservative but a typical value in commercial battery packs, including the safety casing [4]. The factor of 0.85 accounts for the losses in battery and transmission as well as the power required by onboard systems other than motors.

3.2 Power Consumption

3.2.1 Power in hover and vertical flight

Based on the momentum theory, the shaft power required by each rotor is,

$$P_{\text{hover}} = \frac{1}{\eta_{\text{hover}}}\frac{T^{1.5}}{\sqrt{2\rho A}}\ , \tag{10}$$

where T is the thrust of each rotor, ρ is the air density, and A is the rotor disk area. The trim condition in hover yields $T = W_{\text{total}}/n_{\text{rotor}}$. We assume the hovering figure of merit of $\eta_{\text{hover}} = 0.75$ for hexarotors and 0.65 for QBiTs; we set lower efficiency for the QBiTs because we assume that the QBiT rotor (propeller) design is tailored for an efficient cruise at the sacrifice of hover efficiency. The hexarotor's rotor operates in similar conditions in hover and edgewise forward flight, which allows it to have hover-efficient rotors.

3.2.2 Power in cruise

In forward flight, we use the relation based on the momentum theory to compute the power required as follows [33]:

$$P_{\text{cruise}} = TV_\infty \sin\beta + \kappa T V_i + P_0\ , \tag{11}$$

where V_∞ is the freestream velocity, V_i is the induced velocity, β is the shaft tilt angle, and P_0 is the profile power. Accordingly, $V_\infty \sin\beta$ is the component of the freestream velocity normal to the rotor disk. Following the method of Govindarajan and Sridharan [4], the induced velocity V_i and induced power factor κ are given by

$$\begin{aligned} V_i &= \lambda \Omega r - V_\infty \sin\beta \ , \\ \kappa &= \min\left(1.15, \frac{1}{\eta_{\text{hover}}} - \frac{\sqrt{2\rho A}}{T^{1.5}} P_0\right) \ , \end{aligned} \tag{12}$$

where the rotor inflow ratio λ is given by solving the inflow equation

$$\lambda = \mu \tan\beta + \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}} \ . \tag{13}$$

The key non-dimensional parameters are defined as follows:

$$\begin{aligned} \mu &= \frac{V_\infty \cos\beta}{\Omega r} \ , \\ C_T &= \frac{T}{\rho A (\Omega r)^2} \ , \\ J &= \frac{V_\infty}{2 n_{\text{rev}} r} \ , \end{aligned} \tag{14}$$

where μ is the rotor advance ratio in edgewise flight, C_T is the thrust coefficient, and J is the propeller advance ratio for the winged cruise of the QBiT.

The profile power is given by the following formula [4]:

$$P_0 = \frac{\sigma C_{d_0}}{8} \left(1 + 4.65\mu^2\right) (\rho A)(\Omega r)^3 \ , \tag{15}$$

where C_{d_0} is the airfoil zero-lift drag coefficient and σ is the rotor solidity; we use $C_{d_0} = 0.012$ and $\sigma = 0.13$ [4].

3.2.3 Trim conditions in cruise

The trim conditions determine the thrust and shaft tilt angle. For the hexarotor without wing,

$$\begin{aligned} T &= \frac{1}{n_{\text{rotor}}} \sqrt{W_{\text{total}}^2 + D_{\text{body}}^2} \ , \\ \beta &= \arctan\left(\frac{D_{\text{body}}}{W_{\text{total}}}\right) \ , \end{aligned} \tag{16}$$

where D_{body} is the body drag, and n_{rotor} is the number of rotors, which is six for the hexarotor. For the body drag estimation, we use the method of Sridharan et al. [2] and Govindarajan and Sridharan [4]. Their method assumes the UAV body to be a cylinder whose radius is 58% of the rotor radius, and the length-to-diameter ratio is 2.5. Then the drag is given by

$$\begin{aligned} D_{\text{body}} &= \frac{1}{2} \rho V_\infty^2 S_b C_{D,b} \ , \\ C_{D,b} &= 0.1 + 0.2 \cos^3\beta \ , \end{aligned} \tag{17}$$

where the body reference area S_b is the cylinder frontal area, i.e., the radius times the length.

The trim conditions for the QBiT with wings are $n_{\text{rotor}} T = D$ and $L = W$. The drag is

$$\begin{aligned} D &= D_{\text{body}} + D_{\text{wing}} \\ &= D_{\text{body}} + \frac{1}{2} \rho V_\infty^2 S_w \left(C_{D_{0,w}} + C_{D_i}\right) \ , \end{aligned} \tag{18}$$

where S_w is the wing area, $C_{D_{0,w}}$ is the zero-lift drag coefficient of the wing (set to 0.01), and C_{D_i} is the induced drag coefficient given by $C_{D_i} = C_L^2 / \pi \text{AR} e$. We assume a fixed aspect ratio of $\text{AR} = 8$ and Oswald efficiency of $e = 0.8$ [4]. For the body drag, we use the same method as the hexarotor, with the angle of attack of 5 deg (i.e., $\beta = 85\,\text{deg}$) assumed.

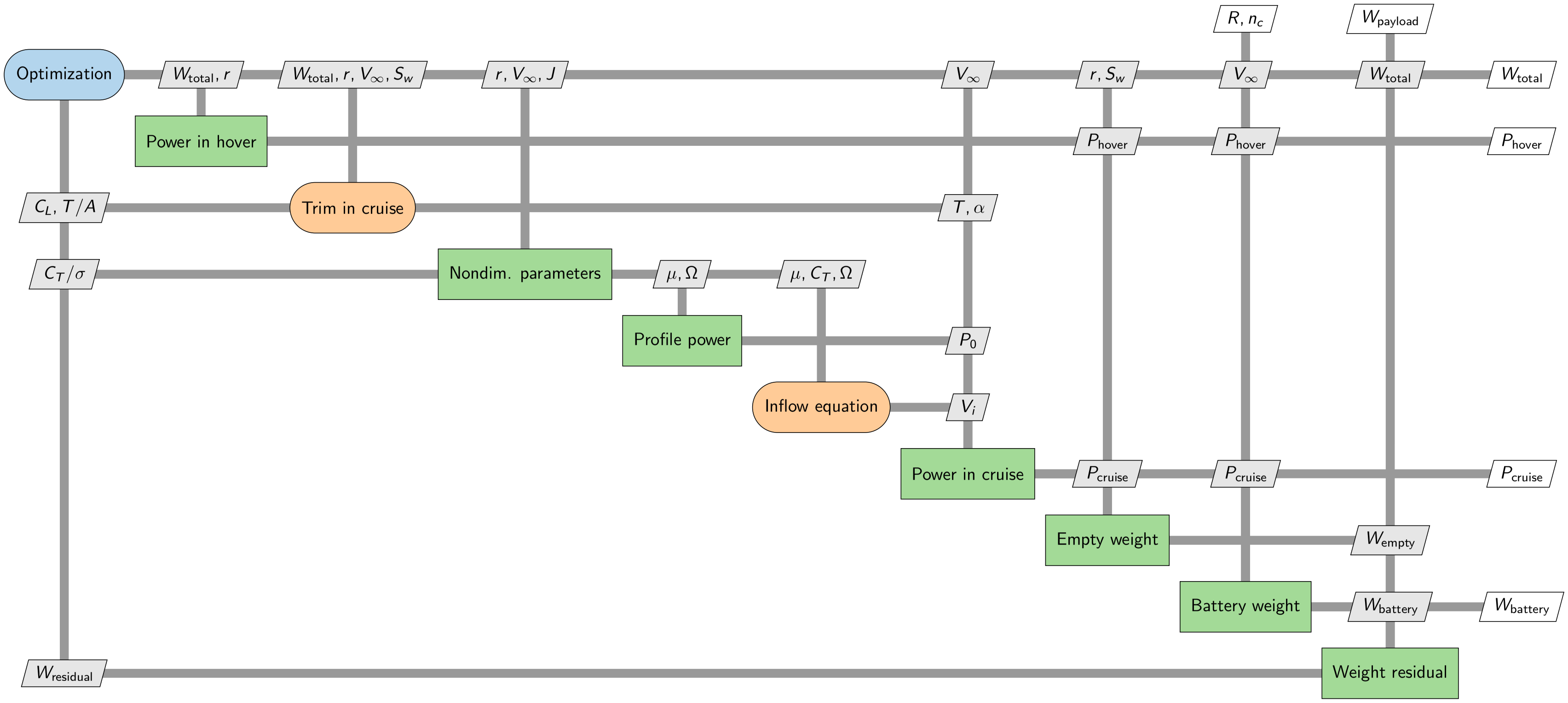


Figure 4: Extended design structure matrix (XDSM) of QBiT sizing optimization.

Table 1: UAV conceptual design optimization problem

	Function/variable	Description	Bounds
Minimize	W_{total}	Takeoff weight, kg	
given	$R, W_{\text{payload}}, n_{\text{c}}$	One or multiple mission requirements	
with respect to	W_{total}	Takeoff weight, kg	[0.5, 50]
	V_∞	Cruise speed, m/s	[10, 50]
	r	Rotor radius , m	[0.05, 1.0]
	μ	Edgewise advance ratio, hexarotor only	[0.01, 0.5]
	J	Propeller advance ratio, QBiT only	[0.01, 1.3]
	S_w	Wing area, m^2, QBiT only	[0.05, 5.0]
subject to	$W_{\text{total}} - W_{\text{payload}} - W_{\text{battery}} - W_{\text{empty}} = 0$	Equality constraint to satisfy Eq. (2)	
	$T/A \leq 250$ N/m^2	Disk loading	
	$C_T/\sigma \leq 0.14$	Blade loading	
	$C_L \leq 0.6$	Cruise lift coefficient, QBiT only	

3.3 Sizing Optimization Problem

Given the mission requirements, we minimize the takeoff weight with respect to the cruise speed, rotor radius, edgewise or propeller advance ratio in cruise, and wing area (only in QBiT cases). The UAV sizing optimization problem is summarized in Table 1. We impose the upper limit on disk loading, blade loading, and the cruise lift coefficient to avoid poor maneuverability and gust response. The upper bound values are the same as those of Govindarajan and Sridharan [4], who explains these constraints.

The sizing model has three implicit variables that need an iterative solver: the rotor inflow ratio λ in Eq. (13); shaft tilt angle β in Eqs. (16) and (17); and takeoff weight W_{total} in Eq. (2). We use a Newton solver for λ and β, whereas we impose the weight residual as an equality constraint instead of solving Eq. (2).

We implemented the sizing models using the OpenMDAO framework [34] with analytic derivatives. We also used the SNOPT [35] optimizer, which is an implementation of the sequential quadratic programming algorithm, via the pyOptSparse wrapper [36].

3.4 Characteristics of Hexarotor and QBiT

This section explains the key characteristics of the UAV models in the context of the design-routing coupled problem. A detailed discussion about the UAV conceptual design and its design space is provided by Sridharan et al. [2], Bershadsky et al. [5] and Govindarajan and Sridharan [4].

The most crucial distinction between the hexarotor and QBiT is their different suitability in various mission settings. The hexarotor is more efficient in hover than the QBiT. Therefore, it is preferable for short-range flights and multi-customer delivery. The QBiT is more efficient in cruise because of its wings, making it a viable option for long-range delivery with a few customers.

Figure 5 shows the optimized takeoff weight of each configuration for single- and 4-customer delivery with various payload weights and mission ranges. The takeoff weight was smaller with hexarotors for shorter ranges, whereas the hexarotors became heavier than QBiTs on longer-range missions. For single-customer delivery, the "switch" of the takeoff weight occurred between 30 and 50 km range, depending on the payload weight. For 4-customer delivery, the hexarotor weight was smaller for all payload weights up to 55 km range, whereas the weight difference between the hexarotor and QBiT became smaller for longer-range missions. This is because the hexarotor is less efficient in long-range missions than the QBiT. As we increase the number of customers, the total hovering time for a mission becomes longer. This implies that the hover performance becomes a more dominant factor than the cruise performance in deciding the optimal design. In other words, the optimal design is tailored for better hover performance when the number of customers is large; the design is tailored for better cruise performance when the number of customers is small or it serves long-range delivery.

Figure 6 shows the power consumption, cruise speed, and rotor radius of the optimized UAVs with a single customer with 3 kg payload. The QBiT consistently required more power in hover than the hexarotor because of its smaller rotor radius, fewer rotors, and lower hover figure of merit. In cruise, the QBiT consumed less power than the hexarotor when the mission range was longer than 43 km, which led to the

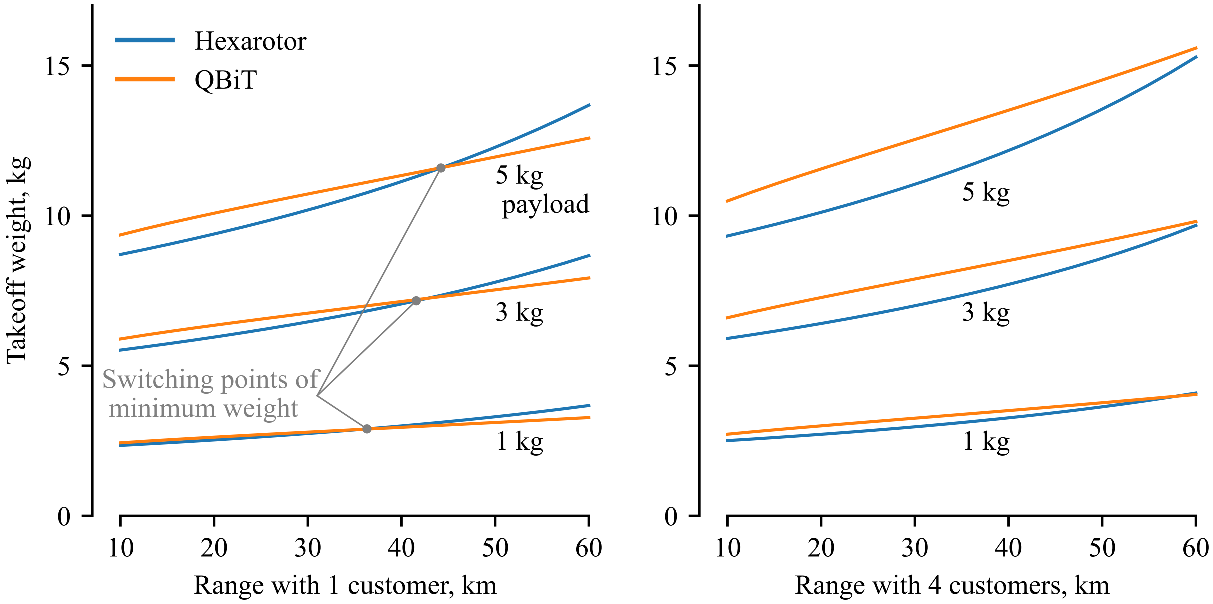


Figure 5: Optimized takeoff weight of the hexarotor and QBiT under various mission settings.

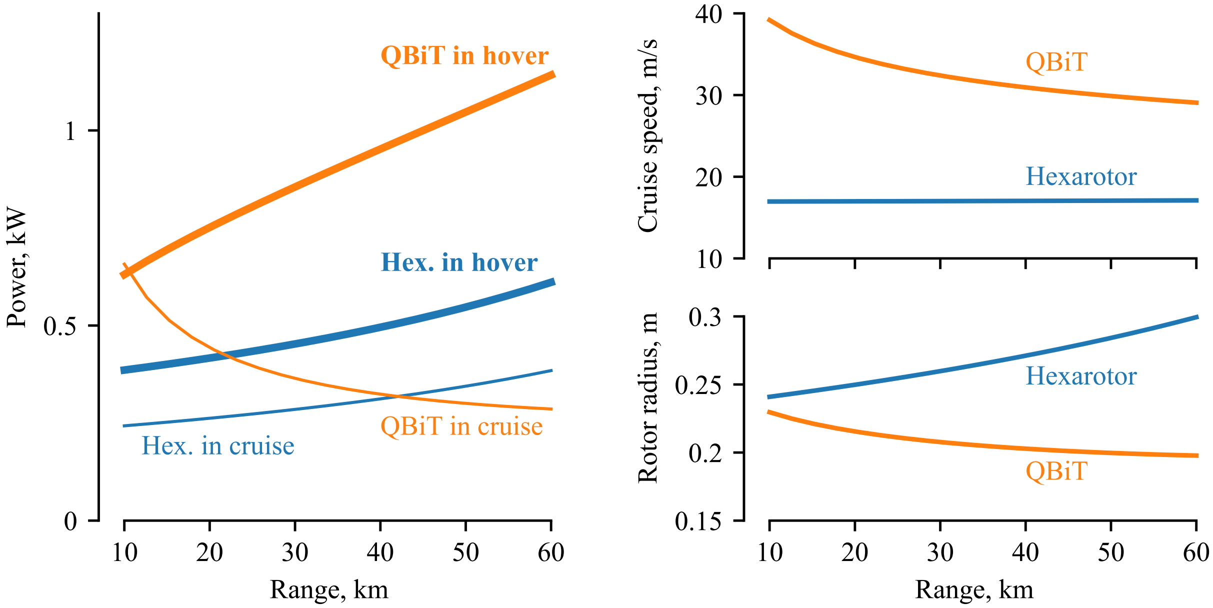


Figure 6: Optimized design variables and power performance for single-customer missions with 3 kg payload. In the power plot, the thicker lines represent the hover conditions, and the thinner lines represent cruise.

smaller takeoff weight, as shown in Fig. 5. In cruise, the wings achieve a low thrust-to-weight ratio, reducing the cruise power.

For the hexarotor, the optimized rotor radius increased as the UAV's total weight increased for longer ranges, but the cruise speed was nearly constant regardless of the range. The optimal speed implies the best balance between the two competing factors: (1) the lower cruise speed results in a longer cruise duration, which increases energy consumption; (2) the lower speed reduces the profile power because the profile power is proportional to V_∞^3 under the constant advance ratio, as shown by Eqs. (14) and (15). The optimal advance ratio was constant because the blade-loading constraint always constrained it; the detail of the blade-loading constraint is discussed by Govindarajan and Sridharan [4]. The lower power consumption in cruise than in hover is mainly because of the translational lift.

The optimal cruise speed and power performance of the QBiT were more sensitive to the mission range than that of the hexarotor because of a trade-off between hover and cruise performance. For the QBiT configuration, the lower cruise speed reduces the energy required for cruise, thus reducing the battery weight. The cruise energy is reduced because the cruise power reduction achieved by the slower cruise is more significant than the increased cruise duration. At the same time, the lower cruise speed increases the wing weight because it requires a larger wing area. In longer-range missions, the lower cruise speed was preferable in minimizing the total weight because the battery weight reduction was more significant than the increase of the wing weight. For short-range missions, the optimizer chose the higher cruise speed and smaller wing area because a lighter wing reduces the hover energy, which was more beneficial than reducing the cruise energy by equipping a larger wing and flying slower.

For the rotor radius variable, a smaller radius improves the cruise efficiency by reducing the profile power and the empty weight, whereas in hover, a larger rotor is preferable to reduce the power, according to Eq. (10). As a result of this trade-off, the optimizer chose a smaller rotor radius for long-range missions where the cruise performance is dominant. In contrast, it preferred a larger radius for short-range missions where hovering is more critical than cruise.

The propeller advance ratio always reached its upper bound of $J = 1.3$. Because $J \propto V_\infty$ and $J \propto r^{-1}$ by definition in Eq. (14), the upper bound of J prevented too small a rotor in the long-range missions or too high a cruise speed in the short-range missions.

3.5 Selection of Objective Function

In this work, the optimization objective is to minimize the sum of takeoff weights of active UAVs in the fleet. The objective function is given by Eq. (19). We intend to reduce fleet acquisition and operating costs by minimizing this objective function. This section discusses the objective function selection and its implication.

Figure 7 shows the difference in the UAV designs and performance between two objective functions. One is the minimum-weight design, and the other is the minimum-energy design that minimizes the total energy required to fly the given mission. Here, we assume that the acquisition cost is proportional to the UAV takeoff weight, and the operating cost is correlated closely to the energy consumption. For the hexarotor, the energy consumption of the minimum-weight and minimum-energy designs is almost identical. Therefore, we conclude that minimizing the hexarotor weight approximately minimizes both acquisition and operating costs.

For the QBiT, the difference between the two objective functions is noticeable, and minimizing the QBiT weight does not imply minimizing the operating cost. However, a downside of the minimum-energy QBiT design is the significantly lower cruise speed than the minimum-weight design. The lower cruise speed reduces the cruise drag and hence the energy consumption at the sacrifice of increased wing area and weight. This result is not preferable, considering that UAVs are expected to enable rapid delivery. This is particularly true for the winged QbiTs. To avoid the low cruise speed, we exclude the energy consumption from the objective function and only minimize the UAV takeoff weight in the remainder of this paper.

3.6 Linear Surrogate Models

The sizing model, which is a mapping from the mission inputs to the sizing outputs as summarized in Eq. (1), is nonlinear because the design optimization performed as a part of the mapping assesses the nonlinear requirements as constraints. When solving the coupled design-routing optimization, the heuristic

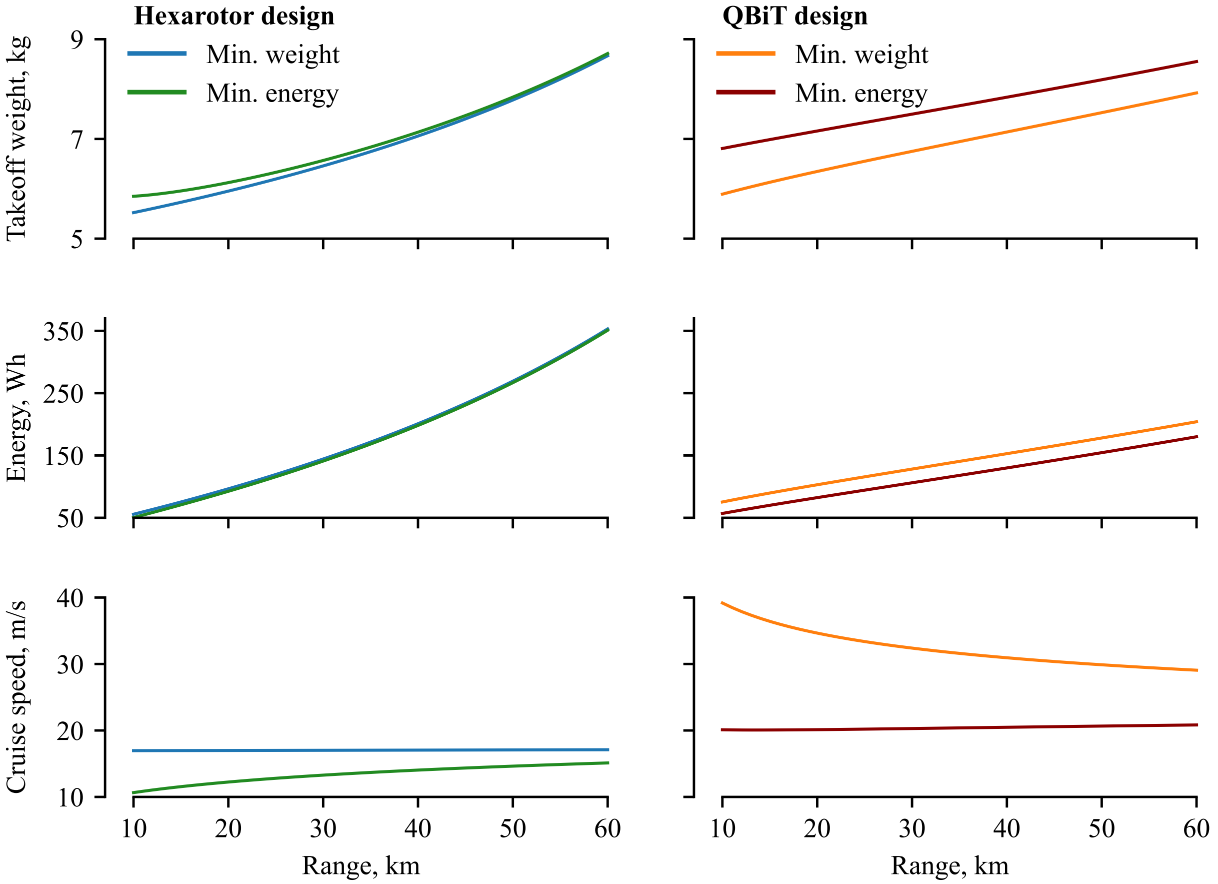


Figure 7: Comparison of the UAV takeoff weight, energy consumption for a mission, and optimal cruise speed between the minimum-weight and minimum-energy designs. We swept the mission range with a 3 kg payload and a single customer. The left column shows the hexarotor results, and the right column shows the QBiT results.

Table 2: Additional constraints and assumptions for design model linearization

Variable	Description	Additional constraint or assumptions to fix the variable
V_∞	Cruise speed	Constant at 18 m/s for hexarotor, 33 m/s for QBiT
r	Rotor radius	Fixed hover disk loading of 120 N/m^2 for hexarotor, 180 N/m^2 for QBiT
μ	Edgewise advance ratio	Fixed at 0.3, hexarotor only
J	Propeller advance ratio	Fixed at 1.3, QBiT only
S_w	Wing area	Constant lift coefficient of 0.6 in cruise, QBiT only

approach (which will be described in Section 4.2) can directly incorporate this nonlinear design optimization. However, the MINLP approach (in Section 4.1) is not compatible with the design nonlinearity. This section builds linear surrogates of the sizing model to enable the coupled optimization by an MINLP branch-and-cut solver.

We first introduce new equality constraints on the disk loading and lift coefficient to determine the rotor radius and wing area. We also fix the cruise speed and advance ratio; the fixed values are summarized in Table 2. The values in Table 2 were based on previous conceptual design optimization results [4]. The sizing optimization now has zero degrees of freedom with these new assumptions.

Next, we train the linear fitting models for W_{total}, P_{hover}, and P_{cruise} in the $(R, W_{\text{payload}}, n_c)$ three-dimensional space. The input domains are $10 \leq R \leq 40$ km, $1 \leq W_{\text{payload}} \leq 10$ kg, and $1 \leq n_c \leq 9$. Here, we set the range upper bound at 40 km to reduce the linearization error by excluding the increasing nonlinearity above 40 km, as shown in Fig. 5. We then build the surrogate models by solving least-square problems with non-negativity constraints on the weight and power predictions. A surrogate for W_{battery} is not necessary because the MINLP formulation directly incorporates the energy capacity constraint. On the nonlinear sizing model, the switching points of the lower-weight UAV configuration were above 40 km range for most missions, as shown in Fig. 5. After linearization, the switching points shifted toward the shorter ranges (typically below 40 km), but the overall behavior remains the same: the hexarotor is lighter for the ranges shorter than the switching point, whereas the QBiT is lighter for the longer ranges. This implies that the design-routing solution with the linear surrogate model still includes both hexarotor and QBiT in the optimal fleet, even though we set the range upper bound to 40 km.

We evaluated the surrogate prediction error using 100 random test points. The average and maximum errors were 6.0% and 24.6% for the hexarotor, and 3.4% and 12.1% for the QBiT, respectively. Although the linear surrogates are not particularly accurate, it is not our primary purpose to have accurate and inexpensive surrogates. The purpose of linearization in this work is to enable the optimization benchmark studies in Section 5, where the MINLP solver is restricted to a linear design model. For benchmarking the optimization approaches, the surrogate accuracy is not crucial as long as we use the consistent design model for both MINLP and heuristic approaches.

4 Optimization Approaches

This section first presents an MINLP formulation and evaluation of the design-operation coupled optimization using a commercial branch-and-cut solver. Because the MINLP solver cannot incorporate nonlinear UAV design models, the formulation requires linearized surrogates instead. The MINLP solver is also limited to a small problem size. To address these challenges, in Section 4.2, we propose a novel approach called the sequential heuristic algorithm (SHA) that can efficiently solve large-scale problems and incorporate nonlinear design models.

4.1 Mixed-Integer Nonlinear Programming (MINLP)

The first approach solves the coupled design-routing optimization using a mathematical MINLP solver. This monolithic optimization approach corresponds to the simultaneous analysis and design (SAND) MDO architecture [29]. This approach formulates the coupled optimization as a nonconvex mixed-integer quadratic constrained program (MIQCP) using the three-index vehicle flow model [37] for routing. The approach allows us to consider a heterogeneous UAV fleet. We use the nonconvex branch-and-cut solver by Gurobi 9.5.0 [38],

Table 3: Definitions of the sets, variables, and other parameters for the MINLP formulaiton

Sets	$\mathcal{N}$	Set of customers, $\mathcal{N} = \{1, 2, ..., n\}$
	$\mathcal{S}$	Subset of customers, $\mathcal{S} \subseteq \mathcal{N}$
	$\mathcal{V}$	Set of nodes, which consists of customers and the depot, $\mathcal{V} = \mathcal{N} \bigcup \{o, d\}$
	$\mathcal{A}$	Set of arcs between nodes, $\mathcal{A} = (\mathcal{V} \backslash \{d\}) \times (\mathcal{V} \backslash \{o\})$
	$\mathcal{K}$	Set of all UAVs, $\mathcal{K} = \{1, 2, ..., n_{\text{UAV}}\}$, $\mathcal{K} = \mathcal{K}^{\text{Hexa}} \bigcup \mathcal{K}^{\text{QBiT}}$
	$\mathcal{K}^{\text{Hexa}}$	Set of hexarotors
	$\mathcal{K}^{\text{QBiT}}$	Set of QBiTs
Indices	i, j	Node index, $i, j \in \mathcal{V}$
	k	Vehicle index, $k \in \mathcal{K}$
Variables	x_{ijk}	$x_{ijk} = 1$ if UAV k travels from node i to j, 0 otherwise
	y_{ik}	$y_{ik} = 1$ if node i is served by UAV k, 0 otherwise
	α_k	$\alpha_k = 1$ if UAV k is active (i.e., serves at least one customer), 0 otherwise
	$n_{c,k}$	Number of customers served by UAV k
	R_k	Range flown by UAV k, km
	$W_{\text{total},k}$	Total weight of UAV k, kg
	$W_{\text{payload},k}$	Payload capacity of UAV k, kg
	$P_{\text{hover},k}$	Power consumption in hover by UAV k, W
	$P_{\text{cruise},k}$	Power consumption in cruise by UAV k, W
Others	$n_{c,\text{total}}$	Total number of customers
	n_{UAV}	*Maximum* number of UAVs available for delivery
	o, d	Depot: we consider a single depot but distinguish the start point o and endpoint d
	q_i	Demand by customer i, kg
	d_{ij}	Distance between node i and j, km
	t_{hover}	Hovering time for takeoff or landing, set to 60 s
	$V_{\infty,k}$	Cruise speed of UAV k, set to 18 m/s for hexarotor, 33 m/s for QBiT
	R_{UB}	Upper bound of single-customer range, set to 40 km
	M	Large constant, $M \geq n_{c,\text{total}}$

which uses the bilinear transformation and spatial branching to handle nonconvex quadratic constraints.

The sets, variables, and other constant parameters used in the MINLP formulation are summarized in Table 3. In the MINLP formulation, we assume that each UAV has its own design variables. Therefore, the fleet may have up to $n_{\text{design}} = n_{\text{UAV}}$ different designs of UAVs. Since each UAV serves only one route, the number of routes in the solution equals the number of distinct designs. In FSMVRP, the number of UAVs used in the routing solution is an optimization variable, and n_{UAV} serves as the upper bound of the number of UAVs.

We impose upper bounds on the payload and energy capacity to avoid unrealistically large-sized UAVs. For the energy capacity, we impose an upper bound on the single-customer range, which is the range of a UAV when it serves only one customer, i.e., a round-trip flight from the depot to a customer. This is equivalent to imposing an upper bound to the energy capacity because of Eq. (9). When a UAV is designed for a multi-customer mission, we compute an equivalent single-customer range of the UAV and constrain it. The range for a multi-customer mission is always shorter than the equivalent single-customer range because UAV hovers longer in a multi-customer case. The upper bound value is 60 km for design-routing problems with the nonlinear design model in Section 6. For the problems with the linearized design model in Section 5, we reduced the range upper bound to 40 km to improve the linear surrogate accuracy. The upper bound of the payload capacity is 10 kg for both linear and nonlinear cases. The selection of these upper bound values is based on the QBiT conceptual design literature [2, 39], which reported electric QBiT designs as large as 9.9 kg payload weight and as long as 56 km range.

The MINLP formulation is summarized as follows:

$$\text{minimize} \quad \sum_{k\in\mathcal{K}} \alpha_k W_{\text{total},k}\ , \tag{19}$$

$$\text{w.r.t.} \quad x_{ijk},\ y_{ik},\ \alpha_k \in \{0,1\},\ n_{c,k} \in \mathcal{I} \qquad \forall i,j \in \mathcal{V},\ \forall k \in \mathcal{K}, \tag{20}$$

$$W_{\text{total},k}, W_{\text{payload},k},\ P_{\text{hover},k}, P_{\text{cruise},k},\ R_k \in \mathcal{R} \qquad \forall k \in \mathcal{K}, \tag{21}$$

$$\text{subject to} \quad \sum_{k\in\mathcal{K}} y_{ik} = 1 \qquad \forall i \in \mathcal{N}, \tag{22}$$

$$\sum_{j\in\mathcal{V}\backslash\{i,o\}} x_{ijk} - \sum_{j\in\mathcal{V}\backslash\{i,d\}} x_{jik} = 0 \qquad \forall i \in \mathcal{N},\ \forall k \in \mathcal{K}, \tag{23}$$

$$\sum_{j\in\mathcal{V}\backslash\{i,o\}} x_{ijk} = 1 \qquad i = o,\ \forall k \in \mathcal{K}, \tag{24}$$

$$\sum_{j\in\mathcal{V}\backslash\{i,o\}} x_{ijk} = y_{ik} \qquad \forall i \in \mathcal{V}\backslash\{d\},\ \forall k \in \mathcal{K}, \tag{25}$$

$$\sum_{j\in\mathcal{V}\backslash\{i,d\}} x_{jik} = y_{ik} \qquad i = d,\ \forall k \in \mathcal{K}, \tag{26}$$

$$\sum_{i\in\mathcal{N}} y_{ik} q_i = W_{\text{payload},k} \qquad \forall k \in \mathcal{K}, \tag{27}$$

$$\sum_{i,j\in\mathcal{A}} x_{ijk} d_{ij} = R_k \qquad \forall k \in \mathcal{K}, \tag{28}$$

$$\sum_{i\in\mathcal{N}} y_{ik} = n_{c,k} \qquad \forall k \in \mathcal{K}, \tag{29}$$

$$\sum_{i\in\mathcal{N}} y_{ik} \leq \alpha_k M \qquad \forall k \in \mathcal{K}, \tag{30}$$

$$\sum_{\forall i,j\in\mathcal{S}} x_{ijk} \leq |\mathcal{S}| - 1, \qquad \forall \mathcal{S} \subseteq \mathcal{N},\ \forall k \in \mathcal{K}, \tag{31}$$

$$2(n_{c,k}+1)P_{\text{hover},k}t_{\text{hover}} + P_{\text{cruise},k}\frac{R_k}{V_{\infty,k}} \leq 4P_{\text{hover},k}t_{\text{hover}} + P_{\text{cruise},k}\frac{R_{\text{UB}}}{V_{\infty,k}} \qquad \forall k \in \mathcal{K}, \tag{32}$$

$$W_{\text{total},k} = f_{\text{w}}^{\text{H}}\left(R_k, W_{\text{payload},k}, n_{c,k}\right) \qquad \forall k \in \mathcal{K}^{\text{Hexa}}, \tag{33}$$

$$P_{\text{hover},k} = f_{\text{ph}}^{\text{H}}\left(R_k, W_{\text{payload},k}, n_{c,k}\right) \qquad \forall k \in \mathcal{K}^{\text{Hexa}}, \tag{34}$$

$$P_{\text{cruise},k} = f_{\text{pc}}^{\text{H}}\left(R_k, W_{\text{payload},k}, n_{c,k}\right) \qquad \forall k \in \mathcal{K}^{\text{Hexa}}, \tag{35}$$

$$W_{\text{total},k} = f_{\text{w}}^{\text{Q}}\left(R_k, W_{\text{payload},k}, n_{c,k}\right) \qquad \forall k \in \mathcal{K}^{\text{QBiT}}, \tag{36}$$

$$P_{\text{hover},k} = f_{\text{ph}}^{\text{Q}}\left(R_k, W_{\text{payload},k}, n_{c,k}\right) \qquad \forall k \in \mathcal{K}^{\text{QBiT}}, \tag{37}$$

$$P_{\text{cruise},k} = f_{\text{pc}}^{\text{Q}}\left(R_k, W_{\text{payload},k}, n_{c,k}\right) \qquad \forall k \in \mathcal{K}^{\text{QBiT}}. \tag{38}$$

As a problem setup, we generate a set of customers with known locations and demands q_i, which can be non-uniform. We then compute the distance of each arc d_{ij} based on the customer locations before running the MINLP solver.

The objective Eq. (19) minimizes the summation of the active UAV weights, which we assume reduces the fleet acquisition cost and operating cost. Constraints Eq. (22) impose that each customer is served by only one vehicle. Constraints Eqs. (23) and (24) ensure the vehicle flow conservations, that is, each route is a closed-loop. Constraints Eqs. (25) and (26) are for the consistency between x_{ijk} and y_{ik}. Constraints Eq. (27) are the vehicle capacity constraints. Constraints Eq. (28) relate the active arcs and the flight range of each route. Constraints Eq. (29) impose the consistency between y_{ik} and $n_{c,k}$. Constraints Eq. (30) let the solver determine α_k from y_{ik}, where M is a large constant (called "big M") that needs to satisfy $M \geq n_{c,\text{total}}$. This big M formulation is valid under condition $W_{\text{total},k} \geq 0$, which is imposed in the surrogate training process for Eq. (33).

Constraints Eq. (31) are the subtour elimination constraints (SECs), where $\mathcal{S}$ is a customer subset. The SECs prohibit subtours, which are the infeasible closed loops that do not include the depot. The number of constraints Eq. (31) is exponential with respect to the number of customers, which is prohibitive for large-scale problems. Therefore, we employ the separation procedure [37] for SECs to avoid imposing an exponential number of constraints. This procedure first solves a relaxation problem without SECs, and every time the optimizer finds a feasible solution to the relaxation problem, it identifies all the subtours in the solution and adds the corresponding SECs. The optimizer then returns to the updated relaxation problem with new SECs, and repeats the process until no subtour is found in the solution.

Constraints Eqs. (32) are the energy capacity constraints. The right-hand side is the energy consumed by vehicle k if it performed a single-customer 40 km-range delivery. The left-hand side is the energy required by vehicle k to serve $n_{c,k}$ customers in the route of distance R_k. The energy constraint requires the energy consumption (left-hand-side) to be less than the maximum energy capacity (right-hand-side). Eq. (32) is linear in conventional routing problems with fixed vehicle designs (i.e., P_hover, P_cruise, and V_∞ are constant). However, in design-routing coupled problems, Eq. (32) becomes nonlinear and nonconvex because P_hover, n_c, P_cruise, R_k, and V_∞ are optimization variables. In other words, the design-routing coupling makes the energy constraints nonconvex. To deal with the nonconvexity, we fixed V_∞ as described in Table 2 and reduced Eq. (32) to bilinear inequalities. The bilinear inequality constraints are still nonconvex, but the Gurobi solver can handle them via the bilinear transformation.

Finally, constraints Eqs. (33)–(38) are the design surrogate models from Section 3.6. These constraints are linear because we use linear surrogates. If we used the original nonlinear design model directly instead, the constraints would become general nonlinear equalities. To the authors' best knowledge, no existing mixed-integer programming solvers, including Gurobi 9.5.0, can practically deal with such general nonlinear constraints.

4.2 Decomposition and Sequential Heuristics

The second optimization approach, a new approach we propose, decomposes the design-routing problem into design optimization and VRP, then solves them sequentially. Such decomposition has been successful in the coupled optimization of aircraft design and network flow [24–26]. Furthermore, we propose a modification to the conventional sequential optimization [23] to avoid local minima and increase the probability of finding a coupled optimal solution.

Figure 8 shows the XDSM of the proposed SHA. The optimization procedure is summarized as follows:

1. Prepare an initial fleet. The fleet is parametrized by variables $\left[W_\text{total}, W_\text{battery}, P_\text{hover}, P_\text{cruise}\right]$ of each UAV.

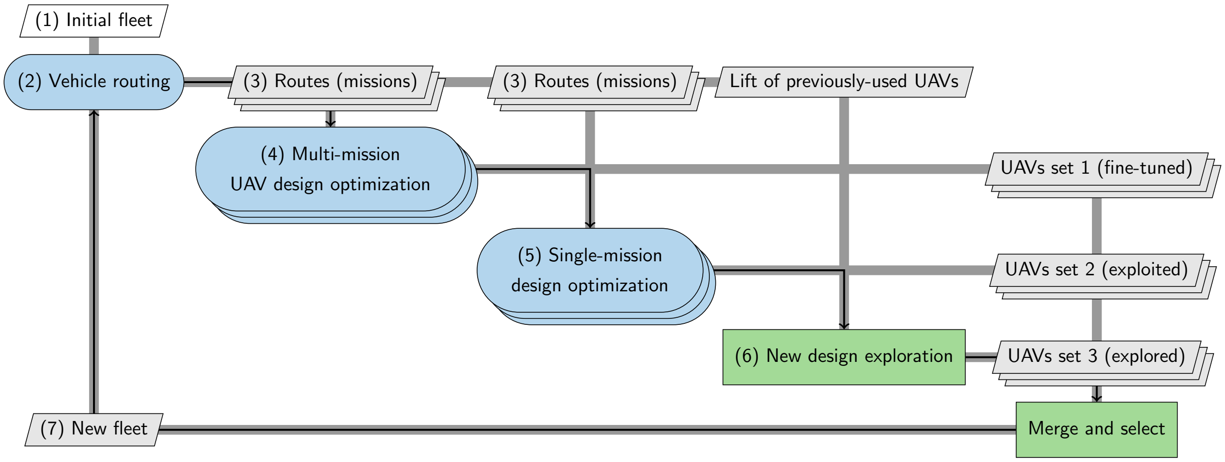


Figure 8: Data flow and processes of the proposed sequential heuristic algorithm (SHA).

2. Solve FSMVRP given the fleet for the minimum fleetwise takeoff weight (Eq. (19)).
3. Given the routing solution, generate a set of flight missions flown by each UAV.
4. Perform multi-mission design optimization of each UAV for the assigned set of missions to fine-tune the designs.
5. Perform single-mission design optimization for each mission (not for a set of missions) to exploit the VRP solution.
6. Introduce exploratory UAV designs to escape from local minima.
7. Repeat Step 2 with the new fleet composed of the UAVs from Steps 4, 5, and 6.

In Step 2, SHA solves FSMVRP using heuristics while fixing the UAV designs in the fleet, i.e., fixing the designs variables W_{total}, W_{battery}, P_{hover}, and P_{cruise}. The VRP here is equivalent to solving Eqs. (19)–(32), whereas it uses a different form of the energy constraints instead of Eq. (32) as follows:

$$2(n_{c,k}+1)P_{\text{hover},k}t_{\text{hover}} + P_{\text{cruise},k}\frac{R_k}{V_{\infty,k}} \leq \rho_b W_{\text{battery},k} \quad \forall k \in \mathcal{K} \ . \tag{39}$$

The left-hand side is the energy consumed, and the right-hand side is the battery energy capacity of the vehicle. SHA employs heuristics to solve VRP because they are the only practical way to handle large-scale problems, given VRP is non-deterministic polynomial-time hard. In the current work, we combine the local cheapest insertion (LCI) algorithm to find an initial feasible solution and the guided local search (GLS) algorithm to escape from local minima and improve the solution globally. Both LCI and GLS are implemented in the Google OR-Tools package [40], and this combination was the most robust and accurate for our FSMVRP among the other algorithms available in Google OR-Tools.

The UAV design optimization in Step 4 achieves same-route improvement, or local improvement, by fine-tuning the design of each UAV for the assigned routes. In designing a fleet, it is not practical to have a distinct specialized design for each route; otherwise, we may have to have hundreds of different UAV designs. Instead, we prefer a fleet with a limited number of designs to reduce the cost of vehicle development and certification. This requires a UAV to be able to serve various missions, i.e., the design requirements for a UAV span multiple missions. Therefore, SHA performs multi-mission design optimization to determine one UAV design given a set of missions assigned to it. SHA uses gradient-based optimization as described in Section 3.3 for the design subproblem. The UAV design optimization subproblem is summarized in Table 1.

In Step 5, SHA performs a series of single-mission design optimization for all missions ignoring the previous assignments by VRP (i.e., regardless of the mission grouping). This step intends to improve the solution by exploiting the routes generated by the previous VRP. For example, suppose two UAVs of design A served route X (20 km, 4 kg payload) and route Y (15 km, 4 kg payload) in the previous VRP solution. Step 4 first optimizes the design of UAV-A while requiring it to fly both routes X and Y; we name the optimized design as Ã. In this simple case, the route X requirements determine design Ã because route X requires more energy than route Y does, which means UAV-Ã will have an extra energy margin when serving route Y. Whereas in Step 5, the proposed method optimizes UAV designs for each route separately, generating new UAV-B for route X and UAV-C for route Y. The design of UAV-B is identical to UAV-Ã, but UAV-C is lighter than UAV-Ã because it only needs to fly the shorter route. Therefore, the objective function would be reduced if the fleet adopts UAV-Ã and UAV-C instead of two UAV-Ãs. However, it may not be feasible to include both UAV-Ã and UAV-C in the fleet when there is a limit on the maximum number of different UAV designs. The VRP solver in the next iteration of SHA will determine whether to adopt the new designs or not. This example comparing Step 4 and Step 5 is illustrated in Fig. 9.

SHA also seeks new-route improvement, which means the global search for better solutions in the design space, in addition to the same-route improvement. Here, we propose a method for new design exploration in Step 6 to achieve global improvement. The procedure of Step 6 is summarized in Fig. 10. In this step, SHA first generates a set of new UAVs whose designs differ from the previously-searched ones. For the design exploration purpose, we represent UAV design by the payload capacity and the range on a single-customer delivery, i.e., $X = [W_{\text{payload}}, R|_{n_c=1}]$, where X is a vector representing the UAV design. SHA then chooses a new exploratory design by repeatedly solving the following optimization subproblem:

$$X_{\text{new}} = \arg\max_{X} \left[\min_{i} \|X - X^{(i)}\|\right] \ , \tag{40}$$

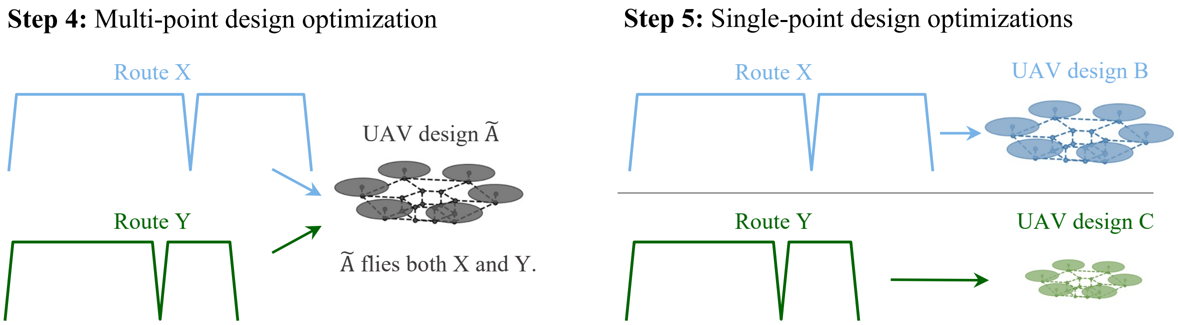

Step 4: Multi-point design optimization
Route X
Route Y
UAV design $\widetilde{\mathrm{A}}$
$\widetilde{\mathrm{A}}$ flies both X and Y.
Step 5: Single-point design optimizations
Route X
UAV design B
Route Y
UAV design C

Figure 9: Example of the multi-mission optimization in Step 4 and single-mission optimizations in Step 5 of SHA.

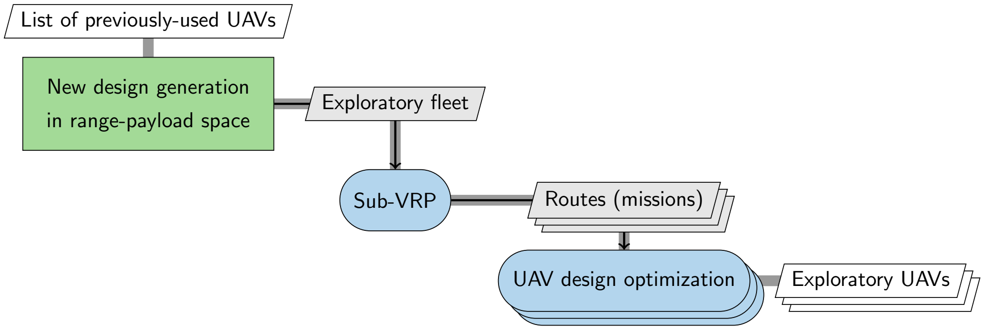


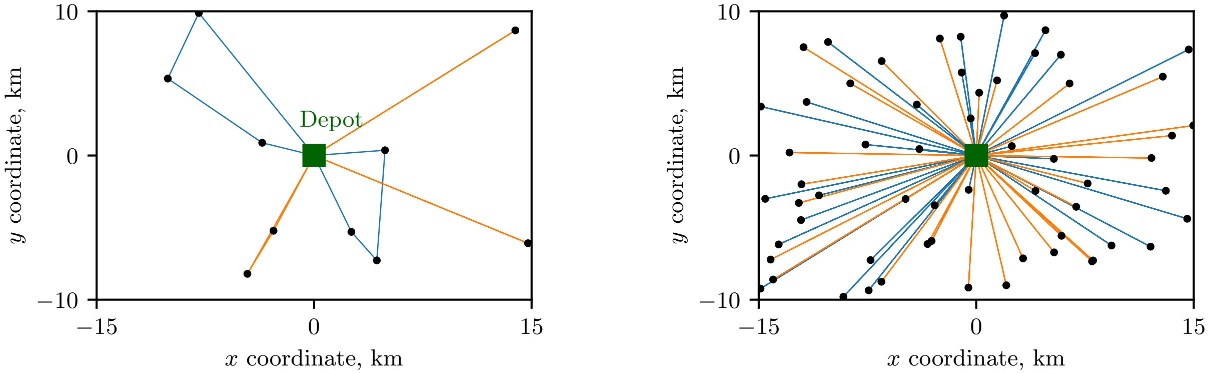
Figure 10: Procedure of new design exploration in SHA.

where $X^{(i)}$ is a previously-used UAV. This subproblem finds the most isolated point from the previously-used designs in the range-payload two-dimensional space. For the UAVs with $n_c > 1$, we compute an equivalent single-customer range, $R|_{n_c=1}$, using Eq. (9) to map the design into the range-payload space. After generating a set of new UAVs, SHA computes $[W_{\text{total}}, W_{\text{battery}}, P_{\text{hover}}, P_{\text{cruise}}]$ for each UAV using Eq. (1) and prepare an exploratory fleet. SHA then solves a sub-VRP using this exploratory fleet. In this sub-VRP, the routing solution does not have to serve all the customers because the exploratory fleet cannot satisfy all the demands. This setting is different from the original VRP in Step 2, where we require the fleet to serve all customers. The exploration step is completed by optimizing the UAV designs for the mission outputs from the sub-VRP.

SHA now has a pool of candidate UAV designs for the next iteration generated from Steps 4, 5, and 6. At this point, SHA may need to select UAVs from the pool when the maximum number of designs is limited. The selection rule, i.e., the portion of the UAVs from each step to be used in the next fleet, is a tunable hyperparameter. In this work, we use the following strategy: in the early sequential iterations, SHA prioritizes the fine-tuned designs as elitism while keeping 30–50% of the spots for exploratory designs, which increases the chance of avoiding local minima in the design space. This strategy is essential for the proposed sequential method to balance exploitation and exploration. The setting of the 30–50% allocation was determined by trial and error on the benchmark problems presented in the next two sections. SHA then reduces the exploration portion and focuses more on local improvement and exploitation toward the end.

5 Benchmark Results of the Optimization Approaches

This section compares the accuracy, robustness, and computational cost of the two optimization approaches proposed in Section 4 by solving a set of benchmark problems. The design-routing optimization problem is summarized in Table 4. In this chapter, we use the linearized surrogate models for UAV sizing due to the limitation of the MINLP solver.



(a) 10 customers and the optimal routes.

(b) 60 customers and the MINLP initial routes.

Figure 11: Examples of the benchmark problem instances. The black points show the customer; the green square at the center is the depot; the blue and orange lines show the routes flown by hexarotors and QBiTs, respectively.

5.1 Benchmark Problems

We created a benchmark set of the design-routing optimization problems with various problem sizes of 5, 10, 15, 30, and 60 customers. We generate 20 problem instances for each problem size by randomly locating the customers in a 30 × 20 km region. Figure 11 shows examples of 10 and 60-customer instances. Each customer has either 1 kg or 2 kg demand in 5, 10, and 15-customer problems, and 0.5 kg or 1 kg demand in 30 and 60-customer problems. The depot is located at the center of the region.

For the fleet settings, the maximum number of UAVs in the fleet is the same as the number of customers. Since the linearized design surrogate was trained based on a single-mission design model and cannot take multi-mission into account, we allow each UAV to have its own design (i.e., the number of different designs is equal to the number of UAVs in the fleet). We applied this condition to both the MINLP and SHA approaches to have a consistent problem setup for the comparison. However, Step 4 (multi-mission design optimization) of SHA is not included in the benchmark study in Section 5; Step 4 will be included in Section 6.

5.2 Accuracy and Robustness of the Sequential Heuristic Algorithm

We first benchmark the proposed sequential heuristic algorithm's accuracy and robustness. We measure the accuracy by computing the errors of the optimized objective value as

$$\text{Objective error} = \frac{f^*_{\text{heuristics}} - f^*_{\text{global}}}{f^*_{\text{global}}} \, , \tag{41}$$

where $f^*_{\text{heuristics}}$ is the objective value optimized by SHA, and f^*_{global} is the proven global optimal value obtained by the Gurobi MINLP solver.

Figure 12 shows the objective function errors in the 5, 10, and 15-customer benchmark problems. A maximum of 15 customers was the practical limit of the problem size on which we can fully converge the MINLP solver. Until converging the optimality gap to less than 1.0×10^{-5}, the solver ran for 48 hours

Table 4: Summary of design-routing optimization problems

		UAV sizing model incorporated	
	Function/Variable	Surrogate (Sec. 5)	Design optimization (Sec. 6)
Minimize	fleetwise takeoff weight ($\sum W_{\text{total}}$)	—	—
with respect to	routing and allocation variables	optimized	optimized
	total number of UAVs in the fleet, n_{UAV}	$n_{\text{UAV}} = n_{\text{design}}$	$n_{\text{design}} \le n_{\text{UAV}} \le n_{\text{UAV, UB}}$
	config., W_{total}, W_{battery}, W_{payload}, P_{hover}, P_{cruise}	optimized	optimized
	V_∞, r, μ or J, S_w	fixed	optimized

in parallel on a 48-core 2.1 GHz computing node for 15-customer problems. We solved each of the 60 test problems from 20 randomly-generated initial fleets; the optimizations ran for 30 sequential iterations for each initial fleet. This multistart strategy from multiple initial fleets increases the probability of finding the global minimum [31, Tip 4.8]. Each translucent dot in Fig. 12 corresponds to the optimized objective value starting from each fleet. The opacity in the plot indicates that multiple optimization runs converged to a similar value: the more frequent runs converged to a point, the more opaque the point becomes.

Most of the dots converged to the zero error on 5-customer problems, which indicates that most optimization runs found the global minimum. In 10-customer problems, most optimization runs converged to the global minimum, whereas some runs resulted in an error of less than 3%. In 15-customer problems, fewer points converged to the global minimum, and most runs resulted in 0–1% errors. The worst-case error in the 15-customer problems was no more than 3%.

Figure 13 shows the probability of finding an acceptable solution within a given tolerance when performing the multistart SHA. We computed the probability of success based on the multistart optimization results shown in Fig. 12. For example, suppose we want to solve a new instance of the 10-customer problem. When we run SHA once, Fig. 13 estimates that the probability of finding the global minimum is 42%. If we compromise and allow an error less than 0.3% with respect to the global minimum, the probability of finding a solution within the 0.3% tolerance is 63%. When we run SHA 10 times from different initial fleets, the probability of finding the global minimum rises to 77%. If we allow an error less than 0.3% and perform ten multistarts, the probability of finding a solution within the 0.3% tolerance is 97%.

To summarize, SHA almost always finds the global minimum in 5-customer problems. The algorithm is also robust on 10-customer problems if we accept an error of 0.3%. The probability of finding the global minimum becomes lower when the number of customers increases to 15. However, it still is reliable in finding a near-optimal solution if we allow an error of 0.3–1.0%. Although it is impossible to validate the accuracy on larger-scale problems because the MINLP solver never converges, these benchmark results demonstrate

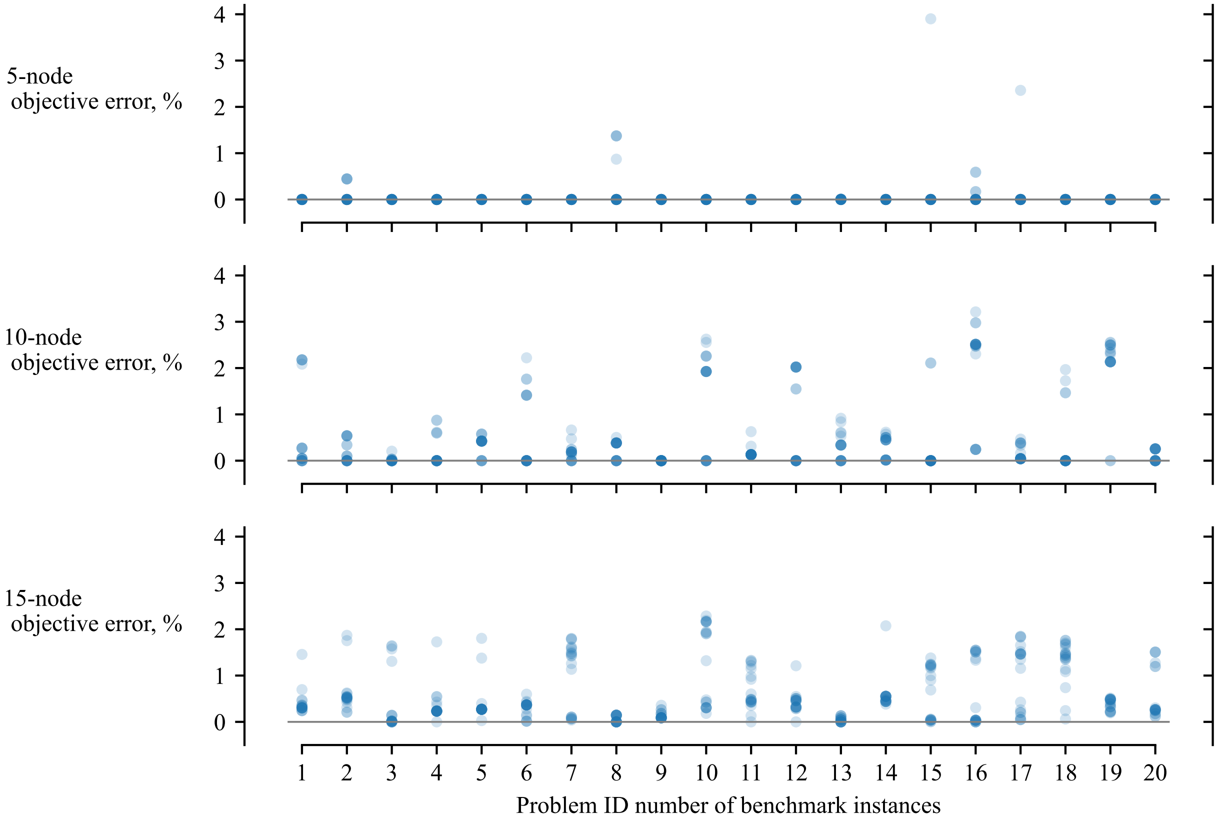


Figure 12: Errors between the global minimum and the objective value found by the multistart SHA.

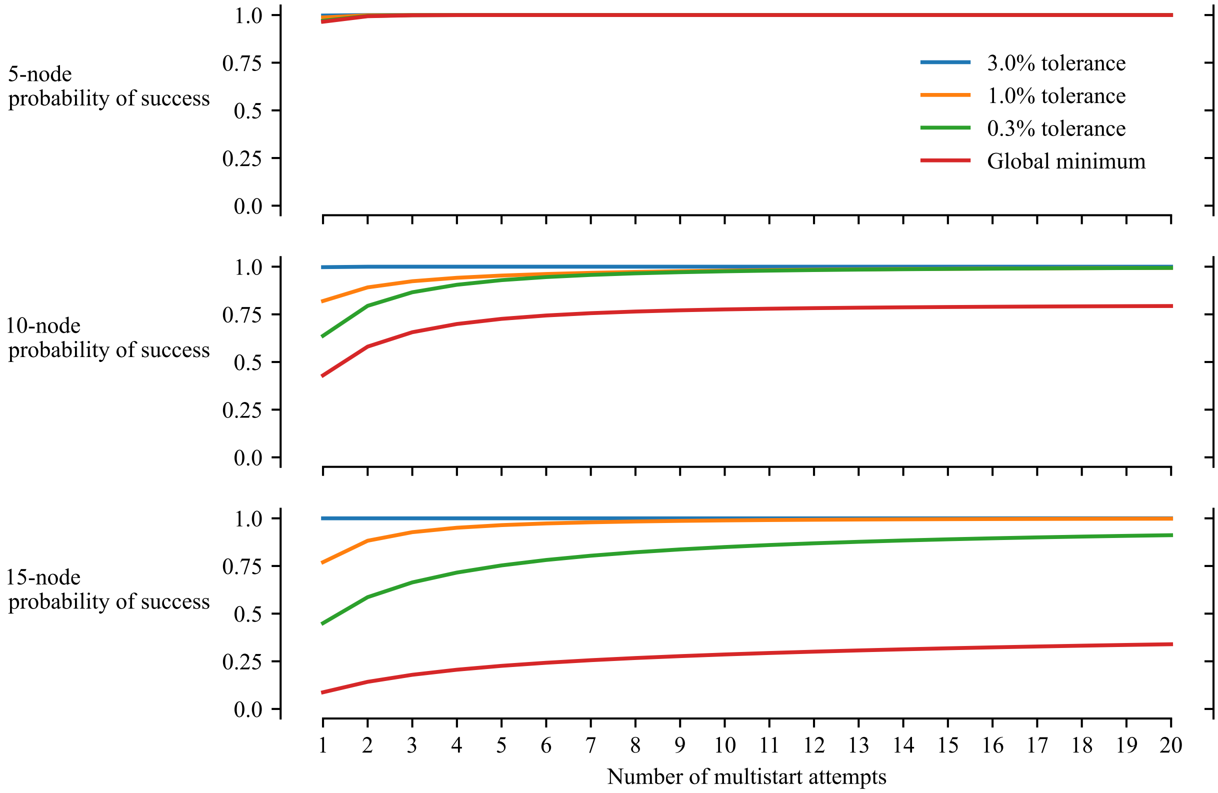


Figure 13: Probability of finding an acceptable solution within a given tolerance when using the multistart SHA.

the accuracy and robustness of the SHA approach we propose.

5.3 Scalability

We also benchmark the scalability of the optimization approaches with respect to the problem size. Here, we solve each benchmark problem by the two approaches under the same amount of computational resources and compare the best solutions found within the given resource. Table 5 lists the wall time limit for each problem size. For a comparison, we run both SHA and Gurobi's MINLP solver on a single thread because the VRP solver by Google OR-Tools does not support parallelization. Table 5 summarizes the resource allocation strategy for SHA, i.e., the number of multistart and sequential iterations that can be done within the time limit. Since the MINLP solver does not benefit from a multistart, we run a single optimization until the time limit. For the 30- and 60-customer problems, we provide a manual initialization to the MINLP solver; otherwise, the solver fails to find a feasible solution within the time limit. The initial solution consists of the most straightforward feasible routes where all customers are directly connected to the depot. An example of the initial solutions is shown in Fig. 11b. The MINLP solver fully converged on all 5 and 10-customer problems, whereas it did not converge for 15, 30, and 60-customer problems. The optimality gap achieved within the given time limit was 6.1–10.4%, 17.6–32.9%, and 31.1–36.3% for 15, 30, and 60-customer problems, respectively. The non-convergence of these problems does not affect the scalability assessment because we only compare the best objective values regardless of the gap convergence.

To compare the two optimization approaches, we define the objective ratio

$$\text{Objective ratio} = \frac{f^*_{\text{SHA}}}{f^*_{\text{MINLP}}}\ , \tag{42}$$

which is the ratio of the best objective value found by SHA to the best objective value found by the MINLP solver (i.e., incumbent solution). The objective ratio of 1 means the SHA and MINLP solvers found the

Table 5: Scalability benchmark study settings

No. of customers	Max. no. of UAVs	Wall time limit, s	SHA resource allocation	MINLP initialization
5	5	250	5 multistarts, 10 iterations	No
10	10	1,000	10 multistarts, 10 iterations	No
15	15	2,250	10 multistarts, 10 iterations	No
30	30	9,000	15 multistarts, 15 iterations	Manual initialization
60	60	36,000	20 multistart, 20 iterations	Manual initialization

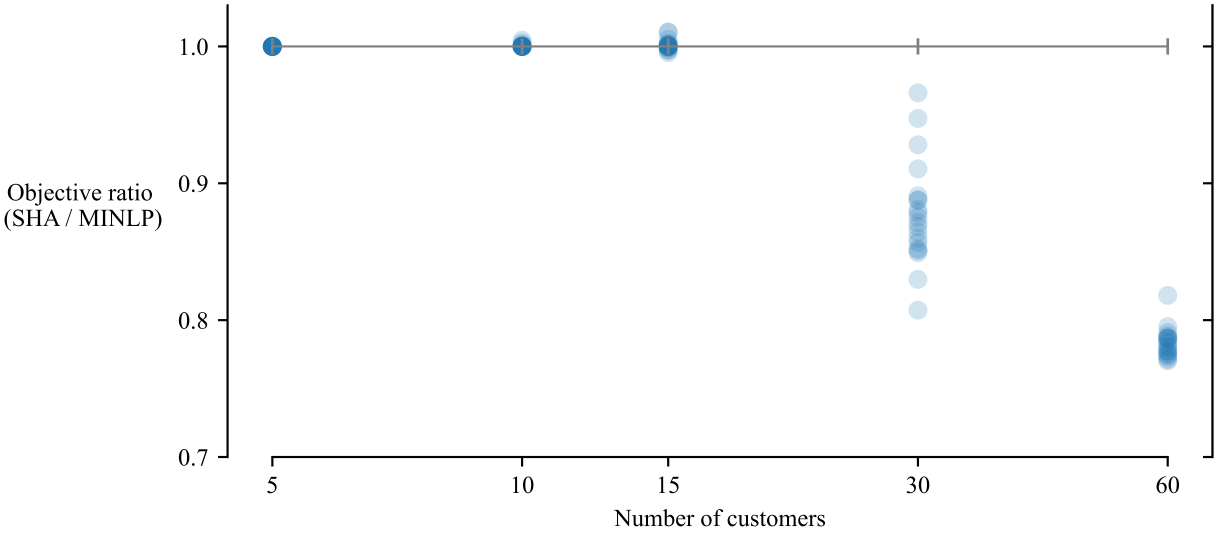


Figure 14: Comparison of the best objective values found by the MINLP solver and SHA approach under the same wall time limit.

same solution. The objective ratio greater than 1 means the MINLP solver finds a better objective value, whereas the ratio less than 1 indicates that the SHA solution is better.

Figure 14 shows the scatter plot of the objective ratio on all 100 benchmark problems. Each translucent dot in Fig. 14 corresponds to each problem. The opacity indicates that multiple benchmark problems resulted in a similar objective ratio. All 20 problems with five customers converged to the objective ratio of 1. On 10 and 15-customer problems, the objective ratio was 1 for most problems. However, the SHA solutions were slightly inferior to the MINLP solution for some problems. These results are consistent with the benchmark results shown in Fig. 12, where we observed 0–3% errors. For 30- and 60-customer problems, SHA always results in a better objective value than the MINLP solution. The difference was 3–20% for 30 customers and 18–23% for 60 customers. The Gurobi MINLP solver showed poor performance even with the manual route initialization because of the NP-hardness of VRP and the nonconvexity of the UAV energy constraints. To summarize, our SHA approach is scalable for large-scale design-routing problems. In contrast, the MINLP branch-and-cut solver is limited to a problem size of 15 customers or less.

6 Results with Nonlinear UAV Design Optimization

This section shows the coupled design-routing optimization results that incorporate the UAV sizing optimization described in Section 3.3. The size of the benchmark problems solved in this section ranges from 5 to 1000 customers. We only use the SHA approach in this section because the MINLP solver cannot handle the nonlinearity of the sizing optimization.

The differences in the optimization setup with and without the UAV sizing optimization are summarized in Table 4. The optimization problem now includes the cruise speed, rotor radius, cruise advance ratio, and wing area as design variables, which were fixed in Section 5. The other difference is that we now limit

Table 6: Representative missions used for the baseline designs

Configuration	Payload, kg	Range, km
Hexarotor	{1, 2, 5, 8, 10}	40
QBiT	{1, 2, 5, 8, 10}	60

the number of different UAV designs to less than the total number of UAVs in the fleet. In practice, it is preferable to have a small number of different designs considering the cost of development, certification, and production. Therefore, we seek to find the optimal fleet with potentially a large number of UAVs but a limited number of different designs.

6.1 Conventional Baseline

We first describe the procedure to obtain a baseline solution, which will be compared to the optimized solution. As a conventional design process, we assume a sequence of design optimization followed by a routing problem with no further iterations. The baseline design process is summarized as follows:

1. Generate n_{design} representative missions.
2. Optimize UAV designs for each mission. This generates n_{design} different designs.
3. Prepare p UAVs of each design and add them to the list of candidate UAVs; the total number of UAVs on the list is the product of p UAVs and n designs, i.e., $n_{\text{UAV}} = pn_{\text{design}}$.
4. Solve VRP to determine the optimal fleet that can serve the customers (i.e., select UAVs out of n_{UAV} candidates such that the summation of takeoff weights across the fleet is minimized).
5. Compute the baseline objective value f_0 by Eq. (19).

In this work, we generate five hexarotors and five QBiT designs, in a total of $n_{\text{design}} = 10$ different designs for all problems regardless of the number of customers. Table 6 summarizes the 10 representative missions we assumed to optimize the designs. We set $p = 2$ for 5, 10, and 15-customer problems. For 30-customer problems or larger, we set p such that the maximum number of UAVs in the fleet is equal to the number of customers, i.e., $n_{\text{c, total}} = n_{\text{UAV}} = pn_{\text{design}}$.

6.2 Comparison of Optimized and Baseline Objective Values

Figure 15 compares the baseline objective value and the optimized values on various sizes of benchmark problems. We solved the same kind of benchmark problems from Section 5, but now we extend the upper bound of the single-customer-trip range to 60 km and the delivery domain to 45 × 30 km. This modification intends to include the "switch point" of the takeoff weight between the hexarotor and QBiT (as shown in Fig. 5), which occurs at 30–50 km depending on the payload weight and number of customers on a route. Furthermore, we also solved larger-scale problems with 120, 250, 500, and 1000 customers. For each size, we prepared 20 problem instances with randomly-generated customer locations. The demand by each customer was the same as Section 5 for problems with 30 or 60 customers; we set a uniform demand of 0.5 kg for problems with 120 or more customers.

SHA repeats the sequential iterations for each problem starting from the baseline fleet until it satisfies the following convergence criteria: 1) a maximum of 100 iterations, and 2) the best objective value is not improved for 20 consecutive iterations. With these criteria, the average wall time required for optimization was 9.8 minutes for a 5-customer problem on a 3.4 GHz CPU, 87.7 minutes for a 60-customer problem on the same CPU, and 83.8 hours for a 1000-customer problem on a 2.1 GHz CPU. All computations were performed on a single thread because the OR-Tools VRP solver (a component of SHA) does not support parallel computation. The VRP solver accounts for most of the computational time: more than 90% of the total wall time was spent on VRP for 60-customer or larger problems. The proportion of the VRP time was less for smaller-scale problems, but it still dominates the overall computational cost. The design optimization of each UAV converges within a few seconds because it is a simple continuous optimization problem solved by a gradient-based optimizer with analytic derivatives.

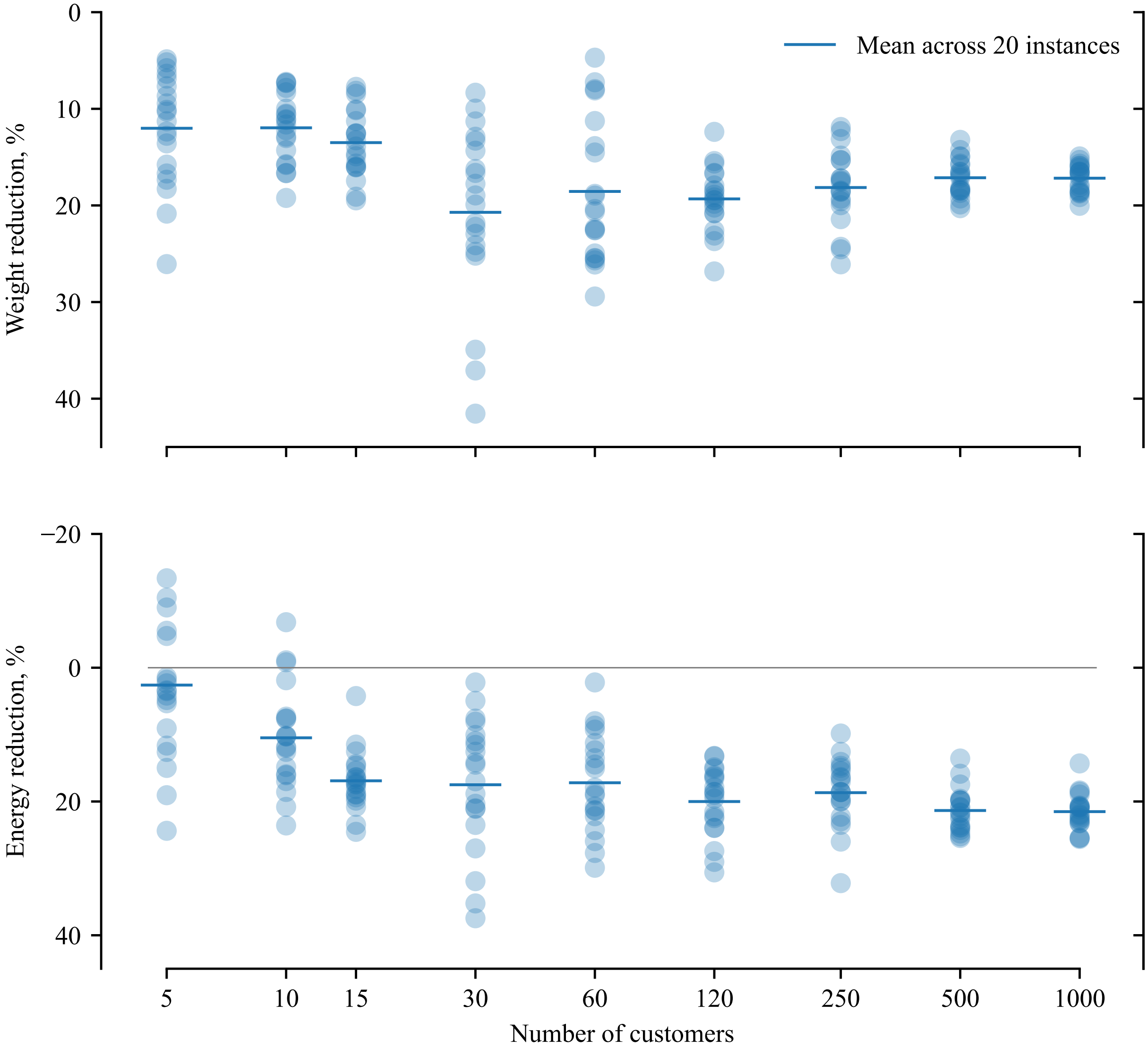


Figure 15: Reduction of the fleetwise summation of takeoff weight (top) and energy consumption (bottom) achieved by optimization compared to the conventional baseline solution.

After convergence, we evaluated the relative improvement of the optimized objective value (summation of takeoff weight across the fleet) with respect to the baseline, given by $|f^* - f_0|/f_0$. We also computed the reduction of the fleetwise total energy consumption between the baseline and optimized solutions. The scatter plots in Fig. 15 show the distribution of the fleet weight and energy reduction for all benchmark problems. The mean improvement of the objective value was 12% or more for all problem sizes. It achieved a 17% mean improvement on large-scale problems with 500 and 1000 customers. We also observed a significant reduction in energy consumption on medium to large-scale problems, even though the objective function does not explicitly include the energy consumption. For 15-customer problems or larger, the mean energy reduction was at least 17%. The reduction of both fleetwise takeoff weight and energy consumption, which implies a lower fleet acquisition cost and operating cost, demonstrates the importance of the design-routing coupling compared to the conventional baseline method.

In Fig. 15, we observe several instances of 5- and 10-customer problems where the weight minimization increased energy consumption. The following two factors caused the energy increase: 1) There is a difference between minimum-weight and minimum-energy designs for QBiT configuration, as discussed in Section 3.5. Therefore, minimizing the weight does not necessarily mean reducing the energy consumption. 2) For small-scale problems, the baseline method may find a "good" initial solution close to the minimum-energy solution. In such cases, minimizing the weight is accompanied by an energy increase. The second factor is only applicable to small-scale problems: On medium to large-scale problems, it is nearly impossible for the baseline method to find such a good initial solution given the problem's complexity. For this reason, the weight minimization always resulted in reducing the energy on problems with 15 and more customers.

6.3 Optimized Routes and Designs

In this section, we illustrate the optimized routes and UAV designs on a 60-customer problem. Although we solved larger-scale problems, we chose the 60-customer problem here for better readability of the figures. This problem has the same customer locations as one of the benchmark problems in Section 6.2, but we modified the demand and UAV fleet settings as follows to facilitate the discussion and visualization: We set a uniform demand of 0.5 kg per customer. The payload capacity of a UAV is obtained by multiplying the number of customers it serves by 0.5. We also reduced the maximum number of different UAV designs to 6 ($n_{\text{design}} = 6$) so that each design is well distinguishable in Figs. 16–18. Correspondingly, we increased the maximum number of the same-design UAVs to 10 ($p = 10$) to allow a maximum of 60 UAVs in the fleet.

Figure 16 shows the optimized routes. Each color corresponds to a different design; multiple routes of the same color indicate that we had multiple UAVs of the same design that served different routes. The blue colors show the routes flown by hexarotors, whereas the red colors represent QBiTs. Figure 17 shows the distribution of the routes in terms of the distance and number of customers along each route. Since the customer demand is uniform at 0.5 kg, the payload weight is obtained by multiplying this demand by the number of customers. The QBiTs served long-range missions with one or two customers, whereas the hexarotors performed 2- to 4-customer delivery with shorter ranges. This trend agrees with the mission suitability of each configuration discussed in Section 3.4.

Figure 18 shows the optimized design variables of all UAVs in the fleet. The optimized fleet has 27 UAVs (13 hexarotors and 14 QBiTs) of six designs. The coordinates in the figure show the UAV takeoff weight and cruise speed, and the UAV illustrations visualize the rotor size and wing area. As we observed in the design-only optimization study in Fig. 6, the hexarotors resulted in larger rotor radii than QBiT, whereas the QBiTs had higher cruise speeds.

On the 2-customer delivery with 28–37 km range, we observe the mixed pattern of the hexarotors (blue circles in Fig. 17) and QBiTs (brown triangles) performing similar missions. Both UAV designs were capable of flying these 28–37 km missions because the hexarotor was designed for the 2-customer 43 km-range mission (the rightmost blue circle) that requires more energy; the same applies to the QBiT, which was designed for the single-customer 49 km-range mission (the rightmost brown triangle). However, the QBiT cannot perform the 2-customer with more than 37 km range because of its inefficiency in hover. The takeoff weight was 3.16 kg for the hexarotor and 3.10 kg for the QBiT; therefore, the optimizer preferred to use the QBiTs as much as possible. As a result, the optimizer used 10 brown QBiTs, which reached the maximum number of the same-design UAVs. Then, the optimizer had to use the hexarotor for the remaining 28–37 km missions, which resulted in 7 hexarotors in total. If we removed the upper bound on p and allowed more than 10 QBiTs, the optimizer would assign QBiTs for all 28–37 km-range missions instead of the heavier hexarotors. In the current study, we imposed this upper bound to limit the number of vehicles, hence the combinations in the routing problem, to keep the problem solvable within a reasonable computational time. Once the designs and number of each UAV were fixed, changing the assignment of UAVs does not change the objective value of the FSMVRP as long as the solution satisfies the energy constraints. In other words, even if we shuffled the assignment of the hexarotors and QBiTs within the 2-customer 28–37 km-range missions, the objective value would remain the same because the objective is the summation of the UAV takeoff weight in the fleet. This explains the mixed pattern of the hexarotor and QBiTs in the 2-customer 28–37 km missions. Additional considerations of cargo volume, delivery speed, and time window may narrow down these assignments. However, the current work does not include these factors in the problem formulation. The UAV assignment discussed here is for the conceptual FSMVRP problem, which is used to determine the optimal fleet design. The assignment for the actual delivery scenarios is determined by solving a more detailed VRP after the fleet is manufactured and day-by-day customer demand becomes available.

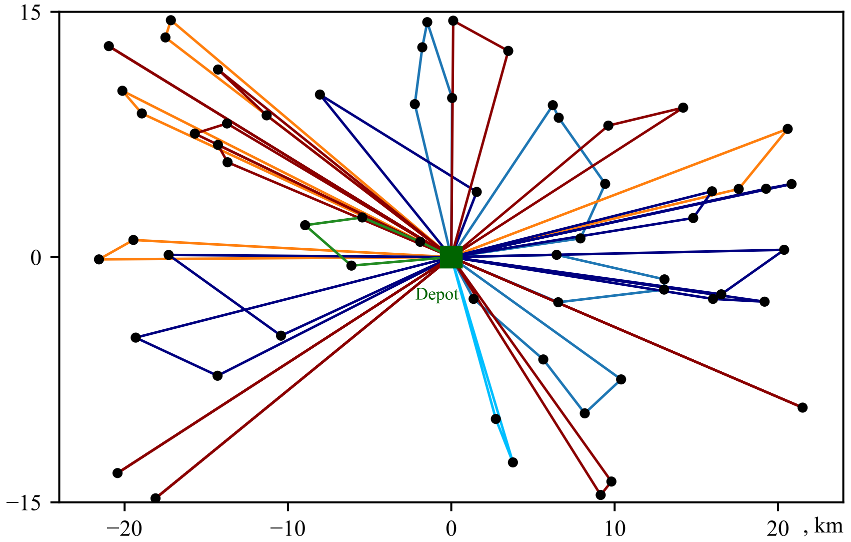


Figure 16: Optimized routes for a 60-customer problem without recharging. The colors correspond to the different UAV designs shown in Fig. 18.

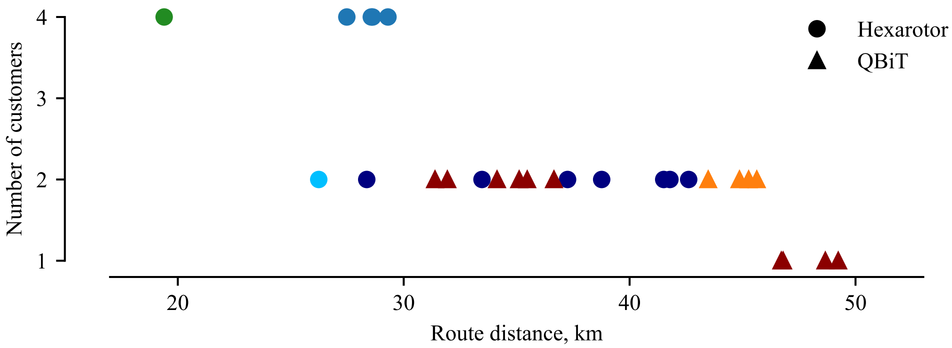


Figure 17: Distribution of the distance and number of customers on each route in Fig. 16

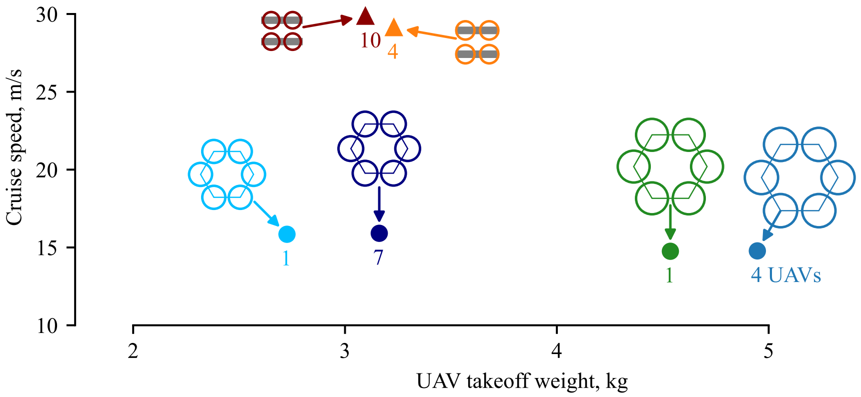


Figure 18: Optimized UAV designs for the problem in Fig. 16. The UAV illustration size scales to the rotor size and wing area variables, and the number label corresponds to the number of each design of UAVs in the fleet.

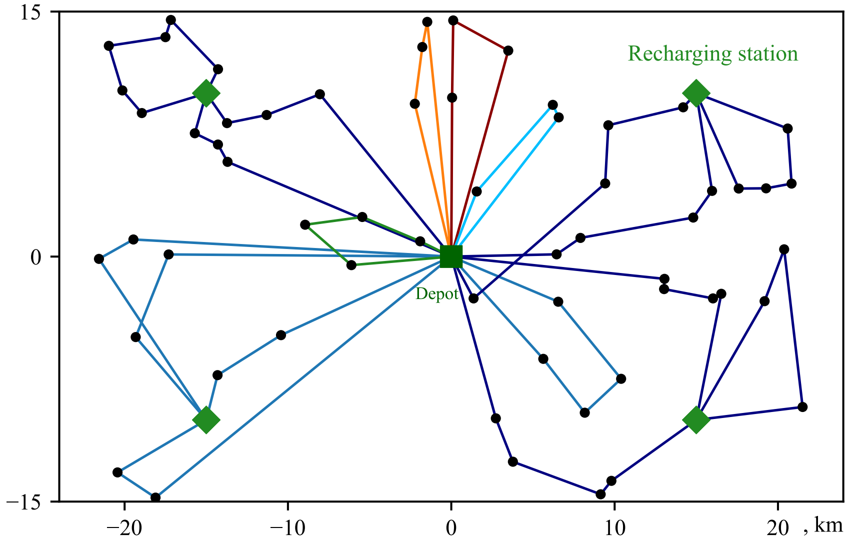


Figure 19: Optimized routes for a 60-customer problem with recharging stations. The colors correspond to the different designs of UAVs shown in Fig. 20.

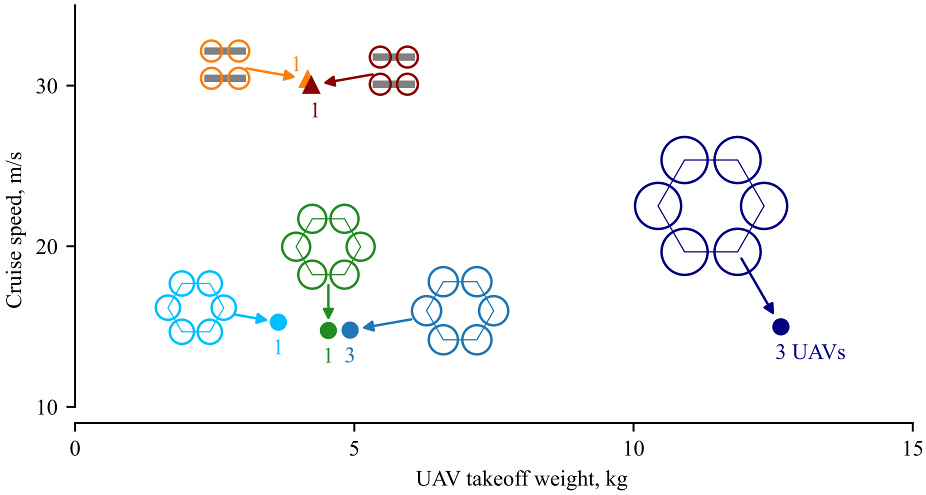


Figure 20: Optimized UAV designs for the recharging problem.

6.4 Solution with Recharging Stations

The modularity of the proposed SHA approach allows us to extend the disciplinary models, i.e., the routing model or conceptual design model, without changing the top-level algorithm. To demonstrate this advantage, we solve a design-routing problem with an improved routing model with battery recharging. Here, we solve for the same customer locations and demands as Fig. 16, and we now add four recharging stations in the region. A UAV can charge its battery state to 100% every time it visits a recharging station to extend the range.

Figures 19 and 20 show the optimized routes and UAV designs for the recharging problem. Compared to the original problem without recharging, the solution with recharging had fewer UAVs in the fleet and introduced larger-capacity hexarotors. The optimized fleet had only 10 UAVs in contrast to 27 UAVs for the problem without recharging; the heaviest UAV was 12.64 kg with recharging, 4.95 kg without recharging. Five of 10 UAVs used the recharging stations; three served 11–12 customers each by recharging twice. The other two UAVs served four customers by recharging once. The other five UAVs served customers far from the recharging stations or near the depot without recharging. The objective function value, the summation of the takeoff weights of all UAVs in the fleet, decreased by 26% compared to the original problem without recharging. This result implies that recharging stations spread out across the delivery domain are effective in downsizing the UAV fleet.

7 Conclusions

In this work, we solved the coupled optimization of UAV conceptual design and delivery operations to reduce the fleet acquisition and operating costs via minimizing the fleetwise takeoff weight. This MDO problem setup enabled us to find a set of optimal mission requirements for which the UAV fleet should be designed and the corresponding optimal UAV designs.

We first presented the conceptual design and sizing model of two eVTOL configurations, the hexarotor and QBiT, and investigated key design characteristics. The QBiT is efficient in cruise because of its wings; therefore, it is suitable for long-range missions with a small number of customers. The hexarotor is efficient in hover and preferable for short-range and multi-customer delivery. We then coupled the eVTOL sizing model to FSMVRP to optimize the UAV fleet design considering delivery operations. We presented an MINLP formulation and solved it using a branch-and-cut solver. The MINLP solver found the global minimum on small-scale problems of up to 15 customers; however, it failed to find a feasible solution for a reasonable computational resource on larger-scale problems. The MINLP formulation was also restrictive in the model fidelity for UAV design, where we had to use the linear surrogate models.

To address these challenges, we proposed the sequential heuristic algorithm (SHA) to solve large-scale problems and incorporate nonlinear design models. The proposed approach combines specialized heuristics for VRP, gradient-based optimization for UAV design, and design exploration strategies to increase the probability of finding the global minimum. We benchmarked the accuracy and scalability of the SHA approach against the MINLP solver on 100 benchmark problems with 5–60 customers. The proposed algorithm found the global minimum in most 5- and 10-customer benchmark problems. It also converged to a near-optimal solution within a 0.3–1.0% error with respect to the global minimum for most of the 15-customer problems. Regarding the scalability with respect to the problem size, SHA consistently found a better solution than the MINLP solver for all 30- and 60-customer problems. We also demonstrated the benefit of incorporating VRP in the conceptual design process of a UAV fleet. Compared to the conventional baseline method, the design-routing optimization reduced the summation of UAV weights across the fleet significantly: We achieved more than 12% mean weight reduction for all problem sizes we tested, including large-scale problems with up to 1000 customers.

In addition to scalability, modularity is another advantage of the proposed SHA approach. Because VRP and UAV design optimization are decomposed, SHA is flexible in model selection and can accommodate a higher-fidelity design model or different VRP formulation as long as the coupling interface remains the same. This paper demonstrated that SHA could switch between two design models, the linear surrogate and the nonlinear sizing optimization, without modifying the top-level algorithm. We also showed the flexibility in the VRP model: the same algorithm solved the design-routing problems with and without battery recharging.

Acknowledgments

This work was partially funded by Israel's Ministry of Defense. The first author was also partially supported by Japan Student Services Organization (JASSO) Fellowship Program. The authors thank Max Opgenoord, Sicheng He, and Tomer Rokita for their feedback on the research methodology. The authors also acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing high-performance computing resources.

References

[1] Stolaroff, J. K., Samaras, C., O'Neill, E. R., Lubers, A., Mitchell, A. S., and Ceperley, D., "Energy Use and Life Cycle Greenhouse Gas Emissions of Drones for Commercial Package Delivery," *Nature Communications*, Vol. 9, No. 409, 2018. doi:10.1038/s41467-017-02411-5.

[2] Sridharan, A., Govindarajan, B., and Chopra, I., "A Scalability Study of the Multirotor Biplane Tailsitter Using Conceptual Sizing," *Journal of the American Helicopter Society*, Vol. 65, No. 1, 2020, pp. 1–18. doi:10.4050/JAHS.65.012009.

[3] Phillips, B., Hrishikeshavan, V., Yeo, D., and Chopra, I., "Flight Performance of a Package Delivery Quadrotor Biplane," *7th AHS Technical Meeting on VTOL Unmanned Air Systems and Autonomy*, 2017.

[4] Govindarajan, B., and Sridharan, A., "Conceptual Sizing of Vertical Lift Package Delivery Platforms," *Journal of Aircraft*, Vol. 57, No. 6, 2020, pp. 1170–1188. doi:10.2514/1.C035805.

[5] Bershadsky, D., Haviland, S., and Johnson, E. N., "Electric Multirotor Propulsion System Sizing for Performance Prediction and Design Optimization," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016. doi:10.2514/6.2016-0581.

[6] Winslow, J., Hrishikeshavan, V., and Chopra, I., "Design Methodology for Small-Scale Unmanned Quadrotors," *Journal of Aircraft*, Vol. 55, No. 3, 2018, pp. 1062–1070. doi:10.2514/1.C034483.

[7] Delbecq, S., Budinger, M., Ochotorena, A., Reysset, A., and Defay, F., "Efficient Sizing and Optimization of Multirotor Drones Based on Scaling Laws and Similarity Models," *Aerospace Science and Technology*, Vol. 102, 2020. doi:10.1016/j.ast.2020.105873.

[8] Coutinho, W. P., Battarra, M., and Fliege, J., "The Unmanned Aerial Vehicle Routing and Trajectory Optimisation Problem, a Taxonomic Review," *Computers & Industrial Engineering*, Vol. 120, 2018, pp. 116–128. doi:10.1016/j.cie.2018.04.037.

[9] Macrina, G., Di Puglia Pugliese, L., Guerriero, F., and Laporte, G., "Drone-aided Routing: A Literature Review," *Transportation Research Part C: Emerging Technologies*, Vol. 120, 2020. doi:10.1016/j.trc.2020.102762.

[10] Poikonen, S., and Campbell, J. F., "Future Directions in Drone Routing Research," *Networks*, Vol. 77, No. 1, 2021, pp. 116–126. doi:10.1002/net.21982.

[11] Ulmer, M., and Thomas, B., "Same-Day Delivery with a Heterogeneous Fleet of Drones and Vehicles," *Networks*, Vol. 72, No. 4, 2018, pp. 485–505. doi:10.1002/net.21855.

[12] Chiang, W.-C., Li, Y., Shang, J., and Urban, T. L., "Impact of Drone Delivery on Sustainability and Cost: Realizing the UAV Potential through Vehicle Routing Optimization," *Applied Energy*, Vol. 242, 2019, pp. 1164–1175. doi:10.1016/j.apenergy.2019.03.117.

[13] Murray, C. C., and Raj, R., "The Multiple Flying Sidekicks Traveling Salesman Problem: Parcel Delivery with Multiple Drones," *Transportation Research Part C: Emerging Technologies*, Vol. 110, 2020, pp. 368–398. doi:10.1016/j.trc.2019.11.003.

[14] Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S., "Vehicle Routing Problems for Drone Delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 1, 2017, pp. 70–85. doi:10.1109/TSMC.2016.2582745.

[15] Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., Haghnazar K., R., Zuidema, D., Lima, M. S., and da Costa, A. R., "A Multi-objective Green UAV Routing Problem," *Computers & Operations Research*, Vol. 88, 2017, pp. 306–315. doi:10.1016/j.cor.2017.04.011.

[16] Choi, Y., Robertson, B., Choi, Y., and Mavris, D., "A Multi-Trip Vehicle Routing Problem for Small Unmanned Aircraft Systems-Based Urban Delivery," *Journal of Aircraft*, Vol. 56, No. 6, 2019, pp. 2309–2323. doi:10.2514/1.C035473.

[17] Cheng, C., Adulyasak, Y., and Rousseau, L. M., "Drone Routing with Energy Function: Formulation and Exact Algorithm," *Transportation Research Part B: Methodological*, Vol. 139, 2020, pp. 364–387. doi:10.1016/j.trb.2020.06.011.

[18] Zhang, J., Campbell, J. F., Sweeney II, D. C., and Hupman, A. C., "Energy Consumption Models for Delivery Drones: A Comparison and Assessment," *Transportation Research Part D: Transport and Environment*, Vol. 90, 2021. doi:10.1016/j.trd.2020.102668.

[19] Kirschstein, T., "Comparison of Energy Demands of Drone-based and Ground-based Parcel Delivery Services," *Transportation Research Part D: Transport and Environment*, Vol. 78, 2020. doi:10.1016/j.trd.2019.102209.

[20] Golden, B., Assad, A., Levy, L., and Gheysens, F., "The Fleet Size and Mix Vehicle Routing Problem," *Computers & Operations Research*, Vol. 11, No. 1, 1984, pp. 49–66. doi:10.1016/0305-0548(84)90007-8.

[21] Hoff, A., Andersson, H., Christiansen, M., Hasle, G., and Løkketangen, A., "Industrial Aspects and Literature Survey: Fleet Composition and Routing," *Computers & Operations Research*, Vol. 37, No. 12, 2010, pp. 2041–2061. doi:10.1016/j.cor.2010.03.015.

[22] Troudi, A., Addouche, S. A., Dellagi, S., and El Mhamedi, A., "Sizing of the Drone Delivery Fleet Considering Energy Autonomy," *Sustainability*, Vol. 10, No. 9, 2018. doi:10.3390/su10093344.

[23] Choi, Y., "A Framework For Concurrent Design and Route Planning Optimization of Unmanned Aerial Vehicle Based Urban Delivery Systems," Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, August 2019.

[24] Taylor, C., and de Weck, O. L., "Coupled Vehicle Design and Network Flow Optimization for Air Transportation Systems," *Journal of Aircraft*, Vol. 44, No. 5, 2007, pp. 1479–1486. doi:10.2514/1.27320.

[25] Mane, M., Crossley, W. A., and Nusawardhana, "System-of-Systems Inspired Aircraft Sizing and Airline Resource Allocation via Decomposition," *Journal of Aircraft*, Vol. 44, No. 4, 2007, pp. 1222–1235. doi:10.2514/1.26333.

[26] Davendralingam, N., and Crossley, W., "Robust Approach for Concurrent Aircraft Design and Airline Network Design," *Journal of Aircraft*, Vol. 51, No. 6, 2014, pp. 1773–1783. doi:10.2514/1.C032442.

[27] Jansen, P. W., and Perez, R. E., "Coupled Optimization of Aircraft Families and Fleet Allocation for Multiple Markets," *Journal of Aircraft*, Vol. 53, No. 5, 2016, pp. 1485–1504. doi:10.2514/1.C033646.

[28] Hwang, J. T., Jasa, J., and Martins, J. R. R. A., "High-fidelity design-allocation optimization of a commercial aircraft maximizing airline profit," *Journal of Aircraft*, Vol. 56, No. 3, 2019, pp. 1165–1178. doi:10.2514/1.C035082.

[29] Martins, J. R. R. A., and Lambe, A. B., "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, Vol. 51, No. 9, 2013, pp. 2049–2075. doi:10.2514/1.J051895.

[30] Roy, S., Crossley, W. A., Moore, K. T., Gray, J. S., and Martins, J. R. R. A., "Monolithic approach towards next generation aircraft design considering airline operations and economics," *Journal of Aircraft*, Vol. 56, No. 4, 2019, pp. 1565–1576. doi:10.2514/1.C035312.

[31] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, Cambridge, UK, 2021, pp. 146–147, 327–351. doi:10.1017/9781108980647, URL `https://mdobook.github.io`.

[32] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[33] Chauhan, S. S., and Martins, J. R. R. A., "Tilt-wing eVTOL takeoff trajectory optimization," *Journal of Aircraft*, Vol. 57, No. 1, 2020, pp. 93–112. doi:10.2514/1.C035476.

[34] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. doi:10.1007/s00158-019-02211-z.

[35] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. doi:10.1137/S0036144504446096.

[36] Wu, N., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A., "pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems," *Journal of Open Source Software*, Vol. 5, No. 54, 2020, p. 2564. doi:10.21105/joss.02564.

[37] Irnich, S., Toth, P., and Vigo, D., *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed., Society for Industrial and Applied Mathematics, 2014, Chap. 1, pp. 1–33. doi:10.1137/1.9781611973594.ch1.

[38] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," , 2018. URL `http://www.gurobi.com`, [retrieved 3 March 2022].

[39] Sridharan, A., and Govindarajan, B., "A Multidisciplinary Optimization Approach for Sizing Vertical Lift Aircraft," *Journal of the American Helicopter Society*, Vol. 67, No. 2, 2022, pp. 1–15. doi:10.4050/JAHS.67.022004.

[40] Perron, L., and Furnon, V., "Google OR-Tools," , 2019. URL `https://developers.google.com/optimization`, [retrieved 3 March 2022].