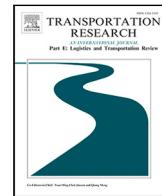




Contents lists available at ScienceDirect

Transportation Research Part E

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



Planning electric vertical takeoff and landing aircraft (eVTOL)-based package delivery with community noise impact considerations

Nahid Parvez Farazi, Bo Zou*

Civil, Materials, and Environmental Engineering, University of Illinois Chicago, United States



ARTICLE INFO

Keywords:

Advanced air mobility (AAM)
eVTOL
Package delivery
Community noise impact
Bi-objective integer programming model
Tailored solution algorithm

ABSTRACT

The rapid development of Advanced Air Mobility (AAM) in recent years suggests a promise to use electric vertical takeoff and landing aircraft (eVTOLs) for package delivery in metro areas. While eVTOL manufacturers and logistics service providers are actively developing prototype eVTOLs and exploring their potentials for moving freight, a system thinking about the suitability and ways to operate an eVTOL-based package delivery system remains scarce. A key aspect of the system thinking is the noise impact of eVTOL operations on surrounding communities. In this study, we provide an operation planning framework that aims to prepare AAM to be both economically efficient and community friendly for package delivery. We first develop a method to quantify the community noise impact of an eVTOL operation, using a “population exposure” measure which is based on the level of sound generated and accounts for both the number of people impacted and duration of the impact. Then, a bi-objective integer programming model is formulated which simultaneously optimizes total shipping cost and community noise impact of eVTOL operations. The optimization takes into consideration operational constraints including maximum distance for local delivery, latest package departure time from the warehouse, and eVTOL fleet size and carrying capacity. A tailored solution algorithm which augments non-dominated sorting genetic algorithm 2 (NSGA2) with compact solution representation, guided generation of initial population of solutions, and customized local search heuristics is devised. The model and the algorithm are implemented in a case study in the Chicago metro area. Numerical results reveal the trade-off between the minimization of shipping cost and community noise impact. Several operational insights about eVTOL-based package delivery are obtained. The computational efficiency and effectiveness of the proposed solution algorithm are also demonstrated in comparison with alternative solution methods.

1. Introduction

Advanced Air Mobility (AAM) is expected to bring revolutionary changes to the transportation system by introducing a new type of aircraft, short named as eVTOLs, that are electric powered and can vertically take off and land (NASA, 2022; Garrow et al., 2021). For freight transportation, eVTOLs offer a promising alternative to ground-based package delivery system, thanks to eVTOLs' faster speed, point-to-point service, and avoidance of traffic congestion on roads. These advantages will only become more prominent given the continued development of online shopping and rapidly increasing demand for same-day delivery (Statista, 2020; Lin et al., 2018;

* Corresponding author.

E-mail address: bzou@uic.edu (B. Zou).

Table 1

Some existing cargo-carrying eVTOL designs.

Sources: FedEx (2022), Klisauskaite (2021), Alcock (2022), Sabrewing Aircraft Company (2021, 2022), Reed (2022), Pipistrel (2022), Garrett-Glaser (2020), Vertical Flight Society (2022).

| | Name | Capacity | Range | Speed | Status |
|---|----------------------|--------------|-----------|---------------|---|
|  | Elroy Air Chaparral | 300–500 lbs | 300 miles | 5×truck speed | Partnering with FedEx in 2022; Scheduled to test middle-mile logistics in 2023. |
|  | Beta Alia 250 | 1400 lbs | 250 miles | 170 mph | Partnering with UPS and Amazon; Tested between Amazon facilities in 2022; Scheduled to launch service in 2024. |
|  | Sabrewing Rhaegal | 5400 lbs | 415 miles | 250 mph | Partnering with US Air Force and Arabian Development and Marketing Company; Deliveries scheduled to start by end of 2023. |
|  | Starling Cargo | 120 lbs | 135 miles | 95 mph | Testing since 2021; Partnering with a South African company; Scheduled to start production by 2022. |
|  | Pipistrel Nuova V300 | 660–1000 lbs | 190 miles | 102 mph | Targeting middle-mile logistics market; Projected to be released in 2023. |

Klein and Steinhardt, 2023). Several eVTOL manufacturers and major logistics service providers (DSPs), such as Amazon, UPS, and FedEx, have already been actively developing prototype eVTOLs and testing their potentials for package delivery (Alcock, 2022; Klisauskaite, 2021; FedEx, 2022). Table 1 shows some of such eVTOL prototypes along with their configurations and development status. Despite these industry advancements, a system thinking about the suitability and ways to operate an eVTOL-based package delivery system remains scarce. Such a system thinking is critical to ensure that this new form of package delivery is not only attractive to the industry, but also acceptable by the public, especially by communities that may be affected by eVTOL operations.

From the perspective of package delivery businesses, eVTOLs will allow DSPs to fly directly from a warehouse to vertiports that are in proximity to final customers. Vertiports are designated areas where eVTOLs vertically take off and land (Heliport Lighting, 2022). From the vertiports, packages are transferred to local transportation modes such as vans, tricycles, and crowdshippers (e.g., Tipagornwong and Figliozzi (2014), Kafle et al. (2017)) to perform the last leg of the delivery. Because takeoff and landing operations are vertical, the land requirement of a vertiport is expected to be small. For example, existing helipads (for helicopter use) are recommended for eVTOL takeoff and landing operations in the early stage of AAM development (FAA, 2020, 2022). As a consequence of the small land requirement, the cost of setting up a new vertiport is projected to be low compared to a delivery station of today, with cost mostly expended in pavement resurfacing, sign marking, and installation of communication equipment.

As opposed to eVTOL-based delivery, conventional delivery relies on ground vehicles to first move packages from the warehouse to intermediate delivery stations (which belongs to “middle mile”), where packages are transferred to smaller vehicles for final delivery to customers (which is “last mile”). Establishing and operating delivery stations incurs a range of costs pertaining to storage space acquisition, building construction, inventory equipment purchase/installation, staffing for package checking, sorting, and transfer, and routine station maintenance. By skipping the intermediate stations, eVTOL-based package delivery straddles middle and last miles, thereby reducing ground vehicle miles in both delivery segments.

From the perspective of community acceptance, an important concern of eVTOL operations is the noise impact generated on surrounding communities. As vertiports are anticipated to be located not far away from final customers, it is likely that eVTOL takeoff and landing operations at vertiports create noise annoyance for people living nearby. Community noise is considered the second top concern after safety for operating eVTOLs in urban areas (Yedavalli and Mooberry, 2019). It also represents a key constraint for community acceptance of scaled eVTOL operations (Vascik and Hansman, 2018).

Conceptually, a trade-off exists between the economic efficiency of eVTOL-based package delivery and its noise impacts on surrounding communities. If minimizing shipping cost is the focus, then package-carrying eVTOLs should choose vertiports close to customers to land. But being close to customers means that the noise impacts on the surrounding areas will be large. On the other hand, if minimizing eVTOL noise impact on communities is of greater interest, then eVTOLs should use vertiports in less populous areas. This however would result in a larger average distance from the used vertiports to final customers, requiring more efforts for local delivery and ultimately increasing total shipping cost. To strike a balance between the economic efficiency and the community noise impact, it is important to first understand the trade-off and how the trade-off is made at the operation planning level.

In view of the above, this paper makes three technical contributions:

- First, we develop a method to quantify the community noise impact of an eVTOL operation, based on the sound level generated and accounts for both the number of people impacted and duration of the impact. A “population exposure” measure is proposed to assess the noise annoyance of an eVTOL flight during vertical ascent and descent to people surrounding the vertiport used by the flight at different times of a day.
- Second, a bi-objective integer program is proposed to simultaneously optimize shipping cost and community noise impact of eVTOL operations. The bi-objective optimization takes into consideration a number of eVTOL operational constraints including maximum distance for local delivery, latest package departure time from the warehouse, and eVTOL fleet size and carrying capacity.
- Third, a tailored algorithm is designed to solve the bi-objective integer program. The algorithm augments non-dominated sorting genetic algorithm 2 (NSGA2) with a compact solution representation, guided generation of initial population of solutions, and customized local search heuristics. The merit of this algorithm is demonstrated by comparing it with several alternative solution methods.

Apart from the above technical contributions, the bi-objective integer program and the tailored solution algorithm are implemented in a case study of package delivery in the Chicago metro area. The numerical results reveal the trade-off between minimizing shipping cost and community noise impact, the eVTOL operational patterns under different objective prioritization, and sensitivity of the Pareto frontier to key parameters for eVTOL operations. The results offer interesting insights about plausible operations of eVTOL-based package delivery. For the rest of the paper, we first review the relevant literature in Section 2. Then, we describe our method for quantifying the community noise impact of an eVTOL operation in Section 3. In Section 4, we formulate the bi-objective integer programming model. Section 5 presents the NSGA2-GG-LS algorithm. Numerical experiments are conducted in Section 6. Section 7 concludes and suggests directions for future research.

2. Literature review

The promise of AAM as part of future transportation has garnered growing interest in recent years (Antcliff et al., 2016; Bulusu, 2019; Rimjha et al., 2021a; Rimjha, 2022; Choi and Park, 2022). Several survey papers have reviewed existing research on various aspects of AAM (Polaczyk et al., 2019; Straubinger et al., 2020; Rajendran and Srinivas, 2020; Garrow et al., 2021). Interested readers may refer to these papers to gain an overview of the recent AAM research. In this section, we confine our focus to two areas closely related to our problem: (1) community noise impact of eVTOL operations, and (2) operation planning of networked AAM systems.

2.1. Community noise impact of eVTOL operations

As mentioned in Section 1, community noise presents an important issue for eVTOL operations (Yedavalli and Mooberry, 2019; Vascik and Hansman, 2018). Addressing this issue hinges on answering two questions: (1) how to quantify the community noise impact, and (2) how to integrate the community noise impact into eVTOL operation planning.

For the first question, several attempts have been made by using the US Federal Aviation Administration’s Aircraft Environmental Design Tool (AEDT) (Rimjha et al., 2021b; Glaab et al., 2019; Jeong et al., 2021). For example, with AEDT, Rimjha et al. (2021b) employ the day-night average sound level (DNL) metric and detailed flight trajectories to quantify noise impact of eVTOL commuting operations for each census block group in northern California and the Dallas-Fort Worth regions. Jeong et al. (2021) resort to the Miedema curve (Civil Aviation Authority, 2018) to estimate the number of annoyed people for Urban Air Mobility (UAM) operations in Seoul, Korea, and subsequently determine the noise priority UAM routes. Considering small Unmanned Aircraft Systems (UAS), Bulusu et al. (2017) quantify the long-term average community noise impact in terms of the affected area and affected population. eVTOL community noise impact has also been quantified based on the acoustics nature of eVTOL sound generation, semi-empirical modeling, and other prediction and simulation tools (Jia and Lee, 2019; Casalino et al., 2019; Schmähle et al., 2022).

For the second question, despite the existing approaches to quantify eVTOL community noise impacts, none of them can be easily integrated in eVTOL operation planning. In fact, the existing approaches are mostly performed assuming fixed flight traffic. As such, they cannot be used in conjunction with decisions on how many, where, and when eVTOL flights are dispatched, which are necessary in eVTOL operation planning. This presents an important research gap, which we intend to fill in the paper.

2.2. Operation planning of networked AAM systems

eVTOL-based package delivery is an understudied subject compared to using eVTOLs for passenger transportation. German et al. (2018) first explore the possibility of using eVTOLs to deliver small packages for the San Francisco Bay Area. By placing a pre-specified maximum number of vertiports to census tracts, the objective is to maximize the demand served by the placed vertiports. However, how to dispatch eVTOL flights towards the vertiports is not part of the decision. German and Daskilewicz (2018) propose four concepts, namely intra-city point-to-point, intra-city hub-and-spoke, regional hub-and-spoke, and city to/from airport, to describe various possible delivery scenarios of eVTOL-based package delivery. Gunady et al. (2022) take a system-of-systems approach to explore the potential of using eVTOLs for middle-mile logistics. A framework for freight demand generation and mode choice in the presence of eVTOLs is developed.

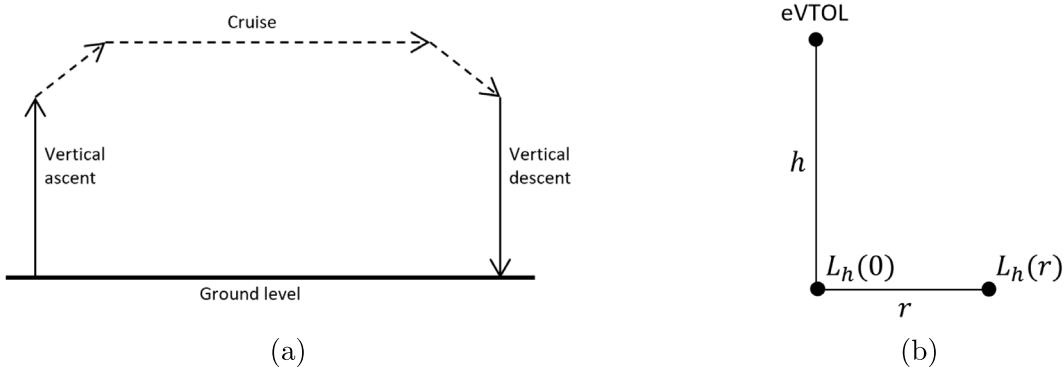


Fig. 1. (a) Illustration of an eVTOL trip. (b) Relative sound levels at directly below and a horizontal distance r .

As opposed to package delivery, AAM operation planning for passenger transportation, which bears some similarities with package delivery, has been researched much more intensively. For example, both package delivery and passenger transportation problems can involve decisions on vertiport selection. Rath and Chow (2022) present a hub location problem to select skyports for air taxi accessing airports. Two objectives are considered: maximizing ridership and maximizing revenue. A similar approach is taken by Willey and Salmon (2021), in which the vertiport location problem is presented as a modified single-allocation p -hub median location problem that incorporates elements of subgraph isomorphism to choose vertiports and vertistops for eVTOL operations. Heuristic algorithms are developed to solve the problem with acceptable computation time and solution quality. Lim and Hwang (2019) use k-mean clustering, Rajendran and Zack (2019) adopt iterative constrained clustering, Daskilewicz et al. (2018) formulate an integer linear program, and Fadhil (2018) employs a GIS-based analysis to investigate vertiport locations for their respective case studies. These studies, however, do not consider eVTOL flight scheduling.

For eVTOL flight scheduling, Bennaceur et al. (2022) group passenger demand into common eVTOL flights that are scheduled to meet expected passenger arrival time at the origin vertiport and potential constraints on arrival time at destination. In doing so, the authors seek to minimize a weighted sum of the number of created flights and passenger waiting time. Ale-Ahmad and Mahmassani (2021) consider pooling passengers who are flexible in where to board and deboard eVTOL flights, and model eVTOL fleet operations as a capacitated location-allocation-routing problem with time windows. A mixed integer program is formulated with decisions on rider request acceptance/rejection, request allocation to flights, and aircraft routing and scheduling. Kim (2019) addresses the on-demand flight scheduling problem with a heterogeneous eVTOL fleet. Particle swarm optimization and genetic algorithm combined with a greedy algorithm are proposed to solve the scheduling problem. Considering a three-vertiport network, Shihab et al. (2019) develop a linear programming model for a hybrid UAM operation concept with both on-demand and pre-scheduled services. Li et al. (2020) investigates a hypothetical UAM network in the San Francisco Bay Area consisting of three vertiports located near the three major airports (SFO, SJC, and OAK) in the area. Performance of the system is evaluated under three scenarios: on-demand operations without fleet re-balancing, on-demand operations with fleet re-balancing, and fixed schedule.

3. eVTOL community noise impact quantification

In this section, we develop a method to quantify the community noise impact of an eVTOL operation. We first characterize the sound level during an eVTOL vertical ascent/descent. Then, we quantify the noise impact of an eVTOL ascent/descent on communities surrounding a vertiport, using a “population exposure” measure.

3.1. Sound level during an eVTOL vertical/ascent at a specific ground location

We consider that an eVTOL flight trip starts with a vertical ascent from the departing vertiport to a certain altitude, followed by an inclined ascent to the cruising altitude, then cruising, inclined descent, and finally vertical descent to the destination vertiport (Fig. 1(a)). The vertical ascent and descent are considered long enough such that the noise impact on the ground, measured by sound level in dB, is below the annoyance threshold level L_a when an eVTOL is at the end of a vertical ascent (or start of a vertical descent). In other words, the noise impact on the ground while an eVTOL is cruising is also below the threshold and can be ignored. Quantifying the eVTOL noise impact thus only needs to look at the vertical ascent and descent parts. A further justification for this is provided when performing the numerical experiments in Section 6.2. Following the literature, L_a is set to 50 dB (Arntzen et al., 2015; Rimjha et al., 2021b; Jia and Lee, 2019; Christian and Cabell, 2017; Glaab et al., 2019).

Before characterizing the sound level during an eVTOL vertical ascent/descent, a few notations need to be introduced. We use $L_h(0)$ to denote the sound level on the ground vertically under a noise source (i.e., an eVTOL) of height h , where 0 means no

horizontal deviation. $L_h(r)$ denotes the sound level on the ground under a noise source of height h with a horizontal distance of r from the noise source. $L_h(0)$ and $L_h(r)$ are illustrated in Fig. 1(b). $I_h(0)$ denotes sound intensity in Watts/m² on the ground vertically under a noise source of height h . $P_h(0)$ denotes sound pressure in Pascal on the ground vertically under a noise source of height h . $I_h(r)$ and $P_h(r)$ are similarly defined with a horizontal distance r .

Expressions for $L_{h'}(0)$ and h' : Because an eVTOL vertical ascent/descent spans a range of altitudes, we want to know $L_{h'}(0)$ and h' for a different height h' given $L_h(0)$ and h . This is done by following the literature Licitra (2012), Bulusu et al. (2017). Specifically, assuming uniform and spherical sound propagation, sound intensity $I_h(0)$ is inversely proportional to the square of height, shown in (1). In addition, $I_h(0)$ is proportional to the square of sound pressure $P_h(0)$, shown in (2). Sound pressure $P_h(0)$ is a function of $L_h(0)$, shown in (3). In (3), P_o is the reference pressure equal to 2×10^{-5} Pascal.

$$I_h(0) \propto \frac{1}{h^2} \quad (1)$$

$$I_h(0) \propto P_h^2(0) \quad (2)$$

$$P_h^2(0) = P_o^2 10^{\frac{L_h(0)}{10}} \quad (3)$$

Using (1)–(3) with some algebra, $L_{h'}(0)$ can be derived as in (4). From (4), h' can be expressed as (5). Details of the derivations are provided in Appendix A.

$$L_{h'}(0) = 10 \lg \left(\left(\frac{h}{h'} \right)^2 10^{L_h(0)/10} \right) \quad (4)$$

$$h' = \left(h^2 (10^{(L_h(0)-L_{h'}(0))/10}) \right)^{\frac{1}{2}} \quad (5)$$

Expression for $L_h(r)$: Given $L_h(0)$, we want to relate $L_h(r)$ to $L_h(0)$. This is not difficult under the assumption of uniform and spherical sound propagation. Specifically, (1) is modified to reflect the distance between the noise source and the impact location, by changing h^2 to $h^2 + r^2$, as shown in (6). (7)–(8) are similarly introduced as they relate to (2)–(3).

$$I_h(r) \propto \frac{1}{h^2 + r^2} \quad (6)$$

$$I_h(r) \propto P_h^2(r) \quad (7)$$

$$P_h^2(r) = P_o^2 10^{\frac{L_h(r)}{10}} \quad (8)$$

Using (1), (2), (6), and (7), $P_h^2(r)$ can be expressed as a function of $P_h^2(0)$:

$$P_h^2(r) = P_h^2(0) \frac{h^2}{h^2 + r^2} \quad (9)$$

Using (8), $L_h(r)$ can also be expressed as a function of $P_h^2(r)$:

$$L_h(r) = 10 \lg \frac{P_h^2(r)}{P_o^2} \quad (10)$$

Substituting $P_h^2(r)$ in (10) by (9) and further substituting $P_h^2(0)$ by (3), $L_h(r)$ can be expressed as a function of $L_h(0)$:

$$L_h(r) = 10 \lg \left(\frac{h^2}{h^2 + r^2} 10^{L_h(0)/10} \right) \quad (11)$$

3.2. Community noise impact of an eVTOL operation

While Section 3.1 quantifies the *instant* sound level at a ground location when an eVTOL is at a height h with a horizontal distance r from the location, the location can be exposed to eVTOL noise for a duration as eVTOL vertically ascents/descents. The duration depends on the horizontal distance r . At one extreme, the location vertically under the eVTOL ($r = 0$) will endure eVTOL noise for the longest time, from takeoff to a height H_n where the sound level at the ground location diminishes to L_a in the case of ascent (and from H_n to touch-down in the case of descent). At the other extreme, the location with the largest horizontal distance from the eVTOL will experience just an instant of sound level L_a , which occurs when the eVTOL leaves ground for ascent (and touches the ground for descent).¹ Under the assumption of uniform and spherical sound propagation, this largest horizontal distance is the same as H_n . Thus, the impacted area on the ground surrounding a vertiport has a circular shape, centered on the vertiport where the eVTOL ascends/descends with a radius of H_n .² For people at different locations in the impacted area, the duration of noise annoyance will be different.

¹ Due to the lack of empirical information, we do not consider the noise impact of an eVTOL on the ground before taking off or after getting landed. In other words, it is assumed that electric motors/propellers are turned on right before takeoff and shut down right after touchdown. Nonetheless, if the empirical information becomes available, the noise impact while an eVTOL is on the ground before takeoff/after touchdown can be easily added in the noise impact calculation.

² The circular shape of the impacted area under the assumption of uniform and spherical sound propagation is reasonable given that our case study area is a suburb (see Section 6) with few tall buildings. On the other hand, the shape of the impact area might need to be adjusted if the case study area were an urban core with high-rise buildings.

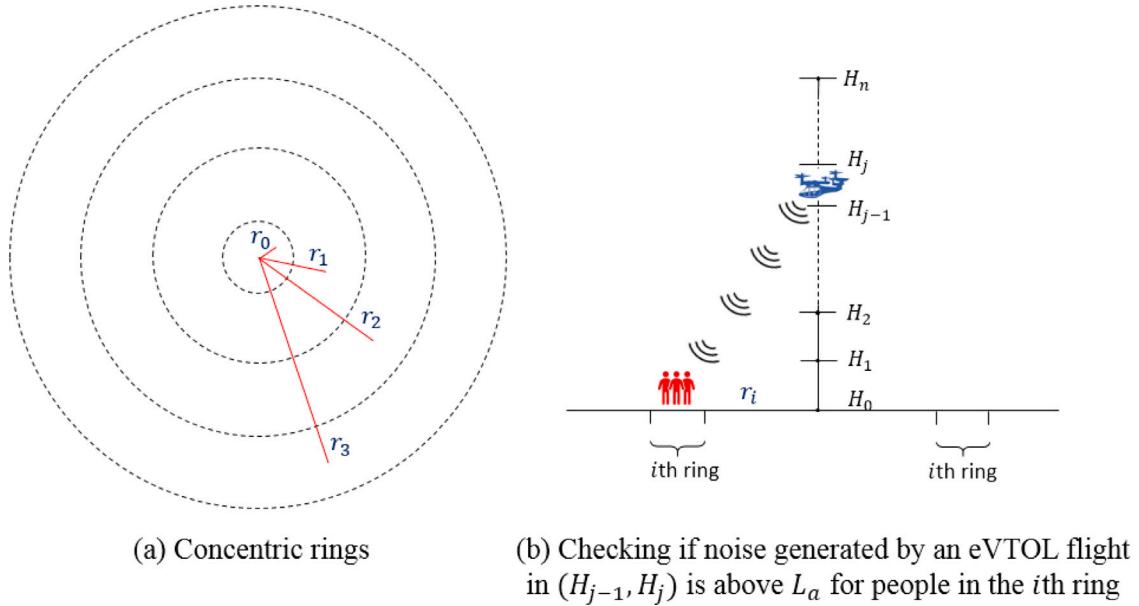


Fig. 2. Illustration of dividing eVTOL noise impacted area into concentric rings and noise impact check.

In view of the above, we quantify the noise impact of an eVTOL operation (including both ascent and descent) at a vertiport by first counting the duration of the noise impact on each person in the impacted area, and then summing the duration across all impacted people. The resulting measure is termed “population exposure” (for simplicity, we also simply use “exposure” later in the paper). The unit is person-minutes. In principle, computing the population exposure requires knowing the location of each person in the impacted area. However, this would be difficult to perform, as we do not have information about the precise location of individual people. Instead, we take a discretization approach.

Specifically, the impacted area surrounding a vertiport is divided into concentric rings of equal width, with the innermost “ring” being a circular area (Fig. 2a). The middle radii of these rings are r_i ’s ($i = 1, 2, \dots$). The height on the ground is denoted as H_0 . The altitudes between H_0 and H_n are divided into n equal-distance intervals delimited by $H_0, H_1, H_2, \dots, H_n$, as shown in Fig. 2b. Given a height interval (H_{j-1}, H_j) and a ring with radius r_i , we check whether the noise generated by an eVTOL flight in the height interval is above L_a for people in the ring. To do so, we use the middle point of the height interval and the middle circle in the ring as references for approximation. The check is performed by plugging $h = (H_{j-1} + H_j)/2$ and $r = r_i$ into (11), to compute $L_h(r)$. If $L_h(r) > L_a$, we count the time an eVTOL spends in the height interval, denoted as T_j , to be the impacting time for people in the ring. T_j is multiplied by the number of people in the ring to obtain the exposure of people in the ring to eVTOL noise while the eVTOL is flying in the height interval. We perform the check for each height interval-ring combination and sum up the exposure over all height intervals and then over all concentric rings, to yield the population exposure to eVTOL noise during an eVTOL ascent (or descent) at the vertiport.

The remaining question is calculating the number of people in each ring during an eVTOL ascent/descent. As our numerical experiments are in the US (see Section 6), this is done by superimposing census block data onto the rings. The census block data, including the shape and population of census blocks, are collected from the [US Census Bureau \(2022\)](#). As a census block can be fully or partially in an impacted area, and can be within one or multiple rings, we calculate the population in each ring by adding up the proportional population from all census blocks that have an overlap with the ring. By doing so, we assume uniform distribution of population in each census block. To be more specific, (12) is used to calculate p_i , the population in ring i surrounding vertiport k .

$$p_i = \sum_{b \in \mathbb{B}_i} \frac{A_b^i}{A_b} p_b \quad \forall i \in \mathbb{R}_k \quad (12)$$

where A_b is the area of census block b , A_b^i the area of census block b that is within ring i , p_b the population of census block b , \mathbb{B}_i denotes the set of census blocks that have an overlap with ring i , and \mathbb{R}_k the set of rings surrounding vertiport k .

Because people are not always at home in a day (e.g., leaving home for work in the morning and returning home in the evening), we apply a fraction parameter ρ_k^t to Eq. (12) to estimate the actual number of impacted people. In this paper, eVTOL dispatching decisions are made at discrete time stamps. As such, ρ_k^t represents the fraction of population in the impacted area surrounding vertiport k , during the operation of an eVTOL flight that is dispatched towards the vertiport at time stamp t . The population exposure to the noise of an eVTOL operation (including both ascent and descent) at a vertiport k for an eVTOL flight dispatched at time stamp

t , denoted as N_k^t , is computed as:

$$N_k^t = 2\rho_k^t \sum_{i \in \mathbb{R}_k} \sum_{j=1}^n p_i T_j 1(i, j) \quad (13)$$

where $1(i, j)$ is a binary indicator of whether the noise generated by the eVTOL flight in height interval (H_{j-1}, H_j) is above L_a for people in ring i . The summation term is multiplied by two to account for both takeoff and landing. This “population exposure” construct is in line with NASA’s interest in assessing noise metrics for single-event eVTOL operations (Rizzi et al., 2020).

As a final note for this section, alternative noise measures also exist, such as day-night average sound level (L_{dn}), community noise equivalent level (L_{den}), and equivalent continuous sound level over a time period T ($L_{Aeq,T}$) based on the A-weighted equivalent continuous sound level (L_{Aeq}). However, computing these measures involves accumulation of sound levels over a period of time (e.g., a day), which would require the entire eVTOL operation plan to be known beforehand. More importantly, as pointed out by Rizzi et al. (2020), the effectiveness of these measures to quantify community noise impact is still unknown. Further exploring these and other possibilities for quantifying community noise impact of eVTOL operations could be an interesting area for future research.

4. Bi-objective optimization of shipping cost and community noise impact

4.1. Preliminaries

In this section, we formulate a bi-objective integer programming model for a DSP to dispatch eVTOL flights, to which packages are assigned, towards a network of vertiports at different times, with the dual objectives of minimizing total shipping cost and community noise impact. To do so, a day of operation is divided into T equal time intervals of length Δ . The beginning of the time intervals is denoted as time stamps $t \in \mathbb{T} = \{1, 2, \dots, |\mathbb{T}|\}$. At each time stamp, the DSP dispatches eVTOLs with loaded packages from the warehouse to vertiports. Upon arriving at a vertiport, packages are unloaded from the eVTOL and continue for local delivery. The DSP has a fleet of E eVTOLs. Each eVTOL has a finite carrying capacity W . Considering the eVTOL flying ranges in Table 1 and the possible size of a metro area, an eVTOL should have no problem making a round trip between the warehouse and any vertiport in a metro area. The set of vertiports is denoted by \mathbb{K} . The set of packages is denoted by \mathbb{J} . For each package $j \in \mathbb{J}$, its final destination, latest delivery time η_j , and weight w_j are known.

After an eVTOL flight is dispatched towards a vertiport k , the amount of time before the eVTOL becomes available consists of five parts: (1) eVTOL outbound travel time from the warehouse to vertiport k , denoted by \mathcal{T}_k ; (2) eVTOL turnaround time at the vertiport (which includes packages unloading from the eVTOL), denoted by ϕ_1 ; (3) eVTOL inbound travel time from the vertiport to the warehouse, which also equals \mathcal{T}_k ; (4) routine eVTOL check and charging after return, denoted by ϕ_2 , and (5) package loading for the next flight, denoted by ϕ_3 . Thus, after an eVTOL is dispatched towards a vertiport k , the eVTOL will need at least $2\mathcal{T}_k + \phi_1 + \phi_2 + \phi_3$ amount of time before the next dispatching. Correspondingly, the number of time intervals during which the eVTOL is unavailable, denoted as γ_k , is:

$$\gamma_k = \left\lfloor \frac{1}{\Delta} (2\mathcal{T}_k + \phi_1 + \phi_2 + \phi_3) \right\rfloor \quad (14)$$

At each time stamp t , we consider that at most one eVTOL flight is dispatched towards a vertiport. In this study, the amount of time between two time stamps (Δ) is much larger than the eVTOL turnaround time at a vertiport (ϕ_1) (see Section 7). Thus, no arrival queue will be formed at vertiports.

The problem of eVTOL-package assignment and eVTOL dispatching is subject to a set of constraints. While Section 4.4 details these constraints, here we conceptually describe two of them. The first constraint relates to what vertiports may be chosen for sending a package. The chosen vertiport should be not too far from the package destination, for the convenience of local delivery. To this end, we impose a maximum distance between the package destination and the vertiport a package is sent to, which is set to be β times the distance between the package destination and its closest vertiport, the latter distance denoted by d_j^* . Similar maximum distance constraints have been considered in other vertiport selection studies for passenger travel as well (German et al., 2018; Daskilewicz et al., 2018).

The second constraint relates to the latest time stamp by which a package needs to be assigned to an eVTOL flight. As each package has a latest delivery time η_j , we consider that when a package leaves the warehouse (i.e., when the eVTOL flight the package is assigned to departs from the warehouse), at least α amount of time should remain. The value of α is set to ensure sufficient time for flying the assigned eVTOL to a vertiport, unloading the packages at the vertiport, and performing local delivery of the package. Supposing that η_0 is the start time of an operation day, the latest time stamp by which a package j needs to leave the warehouse, t_j^l , is:

$$t_j^l = \left\lfloor \frac{1}{\Delta} (\eta_j - \eta_0 - \alpha) \right\rfloor \quad (15)$$

4.2. Decision variables

The integer programming model has two sets of primary decision variables, which are associated with package assignment to eVTOLs and eVTOL flight dispatching:

$$\begin{aligned} x_{jk}^t & \text{ equals 1 if package } j \text{ is assigned to an eVTOL flight that is dispatched at time stamp } t \text{ towards vertiport } k, \text{ and} \\ & 0 \text{ otherwise;} \\ y_k^t & \text{ equals 1 if an eVTOL flight is dispatched towards vertiport } k \text{ at time stamp } t, \text{ and 0 otherwise.} \end{aligned}$$

We consider that violation of the two constraints described above could be tolerated by the DSP, but with penalties (see the objective function specification below). To characterize the violations, two auxiliary decision variables are introduced:

$$\begin{aligned} z_{1,j} & \text{ equals 1 if package } j \text{ is assigned to an eVTOL flight, and thus a vertiport the flight is dispatched towards whose} \\ & \text{distance to the package destination is more than } \beta \text{ times } d_j^*, \text{ and 0 otherwise;} \\ z_{2,j} & \text{ equals 1 if package } j \text{ is assigned to an eVTOL flight which leaves the warehouse later than } t_j^l, \text{ and 0 otherwise.} \end{aligned}$$

4.3. Objective function

As mentioned at the beginning of this section, the bi-objective optimization considers minimizing total shipping cost and community noise impact. The first objective is specified as (16). The first term in (16) denotes the cost of eVTOL operation, where c_k captures the cost of an eVTOL flying a round trip between the warehouse and vertiport k , plus a vertiport usage fee. The second term quantifies the penalty cost associated with $z_{1,j}$'s, where $\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} d_{jk} x_{jk}^t - \beta d_j^*$ denotes the excess distance between the destination of package j and vertiport k to which package j is assigned (d_{jk}) as compared to βd_j^* . θ_1 is the penalty parameter in \$/mile. The third term quantifies the penalty cost associated with $z_{2,j}$'s, where $\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} t x_{jk}^t - t_j^l$ denotes the excess time of package j 's departure from the warehouse with respect to t_j^l . θ_2 is the penalty parameter in \$/time stamp.

$$O_1 = \min \sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} c_k y_k^t + \sum_{j \in \mathbb{J}} \theta_1 \left(\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} d_{jk} x_{jk}^t - \beta d_j^* \right) z_{1,j} + \sum_{j \in \mathbb{J}} \theta_2 \left(\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} t x_{jk}^t - t_j^l \right) z_{2,j} \quad (16)$$

The second objective focuses on minimizing the total community noise impact of eVTOL operations. Building on the population exposure measure of one eVTOL operation (Eq. (13)), the second objective is expressed as (17). In (17), the population exposure of a single eVTOL operation at a vertiport k for an eVTOL flight dispatched towards the vertiport at t (N_k^t) is multiplied by the indicator of whether an eVTOL flight is indeed dispatched towards vertiport k at t (y_k^t). Then we sum the population exposure over all vertiports and all time stamps to obtain the total community noise impact.

$$O_2 = \min \sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} N_k^t y_k^t \quad (17)$$

4.4. Constraints

The optimization is subject to a number of constraints, which are organized in six groups:

(1) package-veriport assignment feasibility:

$$\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} x_{jk}^t = 1 \quad \forall j \in \mathbb{J} \quad (18)$$

$$x_{jk}^t - y_k^t \leq 0 \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (19)$$

(2) eVTOL carrying capacity:

$$\sum_{j \in \mathbb{J}} w_j x_{jk}^t \leq W \quad \forall k \in \mathbb{K}, t \in \mathbb{T} \quad (20)$$

(3) Maximum distance between the chosen vertiport of a package and the package destination:

$$\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} d_{jk} x_{jk}^t - \beta d_j^* \leq z_{1,j} M \quad \forall j \in \mathbb{J} \quad (21)$$

$$\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} d_{jk} x_{jk}^t - \beta d_j^* \geq (z_{1,j} - 1) M \quad \forall j \in \mathbb{J} \quad (22)$$

(4) Latest time stamp for a package to leave warehouse:

$$\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} t x_{jk}^t - t_j^l \leq z_{2,j} M \quad \forall j \in \mathbb{J} \quad (23)$$

$$\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} t x_{jk}^t - t_j^l \geq (z_{2,j} - 1) M \quad \forall j \in \mathbb{J} \quad (24)$$

(5) eVTOL availability at each time stamp:

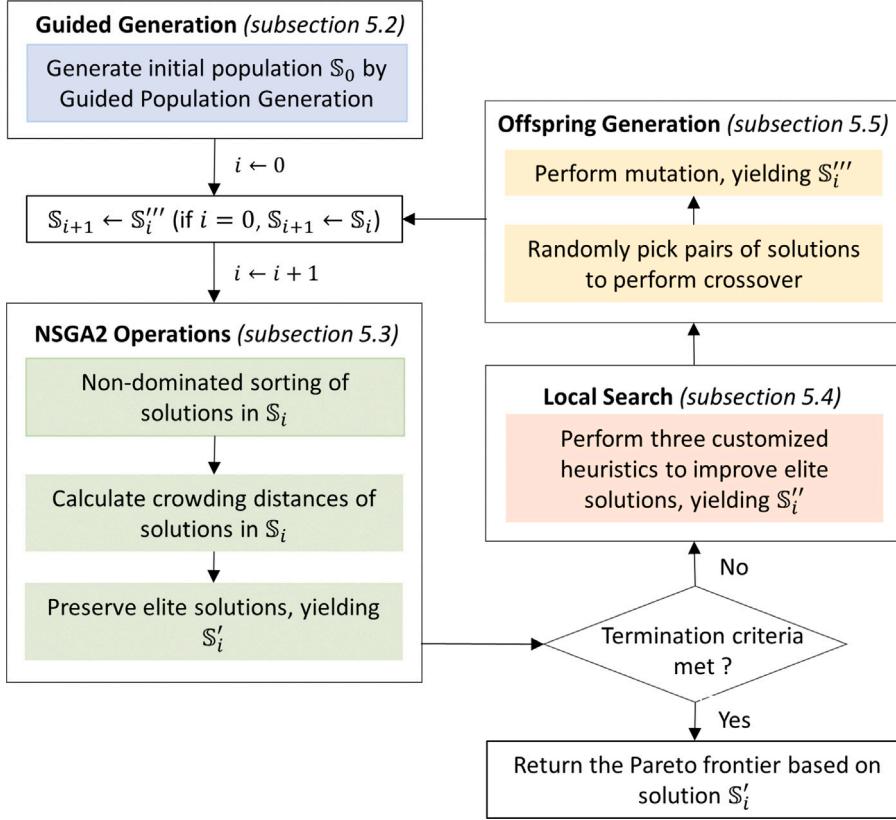


Fig. 3. Overall flow of the NSGA2-GG-LS algorithm.

$$\sum_{k \in \mathbb{K}} y_k^t \leq E \quad t = 1 \quad (25)$$

$$\sum_{k \in \mathbb{K}} y_k^t \leq E - \sum_{i=1}^{t-1} \sum_{k \in \mathbb{K}} y_k^i \bar{\delta}_k^i \quad \forall t \in \mathbb{T}, t > 1 \quad (26)$$

(6) Binary nature of the decision variables:

$$x_{jk}^t = \{0, 1\} \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (27)$$

$$y_k^t = \{0, 1\} \quad \forall k \in \mathbb{K}, t \in \mathbb{T} \quad (28)$$

$$z_{1,j} = \{0, 1\} \quad \forall j \in \mathbb{J} \quad (29)$$

$$z_{2,j} = \{0, 1\} \quad \forall j \in \mathbb{J} \quad (30)$$

For the first group of constraints, constraint (18) specifies that each package is assigned to one and only one eVTOL flight. Recalling that at each time stamp, each dispatched eVTOL flight is associated with a unique vertiport. Thus, the summation is over all vertiports and all time stamps. Constraint (19) ensures that if a package is assigned to an eVTOL flight towards a vertiport at a time stamp, then the assigned eVTOL flight must be dispatched at that time stamp. The second group has only one constraint (20) that the total weight of packages assigned to an eVTOL flight dispatched towards a vertiport does not exceed the carrying capacity of the eVTOL. In the third group, constraints (21)–(22) connect violation of the maximum distance between the chosen vertiport of a package and the package destination, i.e., $\sum_{i \in \mathbb{T}} \sum_{k \in \mathbb{K}} d_{jk} x_{jk}^i - \beta d_j^*$, with the auxiliary decision variable indicating violation of the constraint ($z_{1,j}$) using a big number M . The fourth group of constraints (23)–(24) is similar except that they are about the violation of the latest time stamp by which a package leaves the warehouse.

For the fifth group of constraints, (25) specifies that at the beginning of a day (the first time stamp), all eVTOLs are available. Thus, the number of eVTOL flights dispatched is bounded by the fleet size E . For subsequent time stamps, the number of eVTOL flights that can be dispatched is bounded by the available eVTOLs at the warehouse, considering that some eVTOLs are either in mission or back to the warehouse but going through routine check/recharging and package loading. This is expressed by constraint (26), where a 0–1 indicator $\bar{\delta}_k^i$ is introduced to indicate if an eVTOL dispatched towards vertiport k becomes available again i time

stamps after its dispatch. The value of δ_k^t is determined based on γ_k of Eq. (14). The last group of constraints stipulates that all decision variables take {0, 1} values.

5. Solution algorithm

Because of the trade-off between shipping cost and community noise impact, solutions of the bi-objective integer programming model are not unique. A series of Pareto solutions are expected. To solve the bi-objective model, a tailored, NSGA2-augmented algorithm is devised. In this section, we first propose a compact way to represent the model solutions (Section 5.1). Then, we discuss how our algorithm augments NSGA2, mainly in two aspects: (1) guided generation (GG) of an initial population of solutions (Section 5.2), and (2) customized local search (LS) to improve quality of the elite solutions during iterations (Section 5.4). As a result, the proposed algorithm is termed NSGA2-GG-LS, with “GG” representing “guided generation” and “LS” denoting “local search”. Other operations in NSGA2-GG-LS, namely selecting elite solutions based on non-dominated sorting and crowding distance computing, and crossover and mutation, are briefly described in Sections 5.3 and 5.5.

The overall flow of NSGA2-GG-LS is shown in Fig. 3. The algorithm starts by guided generation of an initial population of solutions, termed \mathbb{S}_0 , and then iterates among elite solution selection (NSGA2 operations), customized local search to improve elite solutions, and crossover and mutation (offspring generation). In Fig. 3, \mathbb{S}_i denotes the population of solutions at the beginning of iteration i . \mathbb{S}'_i , \mathbb{S}''_i , and \mathbb{S}'''_i denote the population of solutions after NSGA2 operations, local search, and offspring generation in iteration i , respectively. The iterations stop after performing a pre-specified maximum number of iterations, or the obtained Pareto frontier does not change in the last three iterations.

5.1. Solution representation

Recall that the decision variables of our model are x_{jk}^t 's and y_k^t 's, both of which are binary. While we could have a binary array of dimension $|\mathbb{J}| \times |\mathbb{K}| \times |\mathbb{T}|$ to represent x_{jk}^t 's and another binary array of dimension $|\mathbb{K}| \times |\mathbb{T}|$ to represent y_k^t 's, given the interconnection between x_{jk}^t 's and y_k^t 's, we propose a more compact solution representation. Specifically, a solution s is represented by two vectors U and V : $s = (U, V)$. U is a vector of length $|\mathbb{J}|$ in which each element U_j is associated with a package j indicating the eVTOL flight to which the package is assigned. Consequently, the number of variables for package-related decisions is reduced from $|\mathbb{J}| \times |\mathbb{K}| \times |\mathbb{T}|$ to $|\mathbb{J}|$. The other vector V provides additional information about eVTOL flight dispatching. Because at most one eVTOL flight is dispatched towards a vertiport at each time stamp, flight dispatching information can be represented by a vector of length $|\mathbb{K}| \times |\mathbb{T}|$, where each 0–1 element indicates whether a flight is dispatched towards a vertiport at a time stamp. The position of a “1” element in the sequence of 1's gives the associated flight number. Given V , we can construct the set of scheduled eVTOL flights $F = \{f_1, f_2, \dots\}$, where each flight is represented by two elements: the vertiport it is dispatched towards and at what time stamp: $f = (k_f, t_f), \forall f \in F$.

Let us use an example to illustrate (Fig. 4). This example corresponds to a problem with $|\mathbb{J}| = 10$, $|\mathbb{K}| = 2$, and $|\mathbb{T}| = 4$. Thus, the length of vector U is 10. Each element in U is associated with a package indicating the flight to which the package is assigned. The length of vector V is $2 \times 4 = 8$. The corresponding total number of variables is $10 + 8 = 18$. In comparison, if we directly use x_{jk}^t 's and y_k^t 's, the total number of variables would be $10 \times 2 \times 4 + 2 \times 4 = 88$, which is much larger. From the example, we can see that $F = \{f_1, f_2, f_3\}$. For each flight, the vertiport it is dispatched towards and the time stamp of dispatching can be found in vector V . The first element of value 1 in V corresponds to flight f_1 . Thus, f_1 is dispatched towards vertiport 1 at time stamp 1, i.e., $f_1 = (1, 1)$. Similarly, we can see that $f_2 = (1, 2)$ and $f_3 = (2, 2)$. In sum, $F = \{(1, 1), (1, 2), (2, 2)\}$.

5.2. Guided generation of initial population of solutions

NSGA2-GG-LS starts with an initial population of solutions, whose size (number of solutions) is pre-specified as $2G$ where G is a positive integer. While a common approach is to start with a randomly generated population (Liu et al., 2016; Karimi-Mamaghan et al., 2020; Guo et al., 2022; Jahani et al., 2022), such an approach may generate initial solutions that are far away from the true Pareto frontier. Instead, we propose a guided process to generate an initial population of solutions to be closer to the Pareto frontier. Starting with such solutions helps reduce the solution search/improvement efforts, leading to more rapid and accurate approximation of the true frontier. Below we first describe how vertiports are ranked, which is the building block for generating one solution. Then we present how the initial population of solutions is generated.

5.2.1. Ranking vertiports by attractiveness

We rank vertiports based on their attractiveness to packages, for which we consider both cost and noise perspectives. To characterize a vertiport k 's attractiveness from the cost perspective, we consider (1) the cost of eVTOL flying a round trip between the warehouse and vertiport k plus vertiport usage fee, which is c_k in the objective O_1 , and (2) the cost associated with distance violation. As we do not know *a priori* which packages will use vertiport k , the distance violation cost is computed over all packages, following the second term in the objective function $O_1: \theta_1(\sum_{j \in \mathbb{J}} \max(0, d_{jk} - \beta d_j^*))$. Note that the first cost occurs to a vertiport k as long as one package uses the vertiport, whereas the second cost to vertiport k accounts for distance violation of all packages. Thus, the two costs may not be added together. Instead, we normalize c_k and $\theta_1(\sum_{j \in \mathbb{J}} \max(0, d_{jk} - \beta d_j^*))$ by min–max scaling based on their respective minimum and maximum values across all vertiports, and then add the two normalized values, to give the cost-side attractiveness of vertiport k , denoted as $A_{k,1}$.

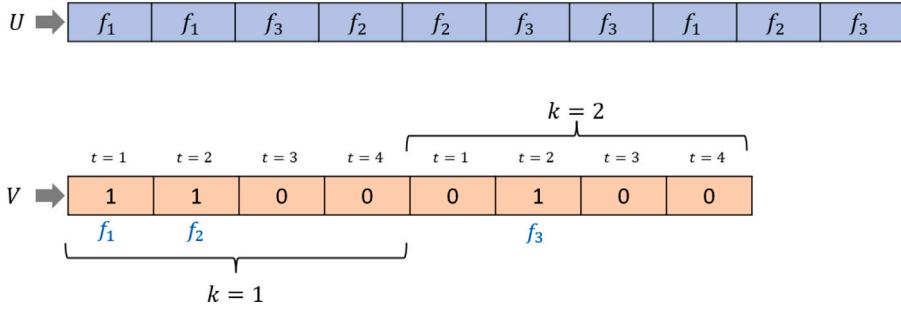


Fig. 4. An illustrative example of solution representation.

To characterize a vertiport k 's attractiveness from the noise perspective, we use the average community noise impact considering dispatching an eVTOL flight to the vertiport at all possible time stamps, i.e., $\sum_i N_k^i / |\mathbb{T}|$. We again normalize this average impact by min–max scaling to obtain the noise-side attractiveness of vertiport k , denoted as $A_{k,2}$. The attractiveness of vertiport k , denoted as A_k , is a weighted sum of $A_{k,1}$ and $A_{k,2}$, weighted by parameter σ and $1 - \sigma$ with $\sigma \in (0, 1)$.

$$A_k = \sigma A_{k,1} + (1 - \sigma) A_{k,2} \quad (31)$$

5.2.2. Generating initial solutions

The ranking of vertiports forms a building block for guided generation of initial solutions. As both cost and noise impact minimization depend critically on the number of eVTOL flights dispatched, the idea of guided initial solution generation is to dispatch a minimum number of eVTOL flights to vertiports to accommodate all packages, by accounting for package weights, eVTOL carrying capacity, eVTOL fleet availability, and vertiport ranking. In doing so, preserving some randomness is also desired to diversify the generated solutions. Algorithm 1 outlines the steps to generate one initial solution.

Algorithm 1 Initial solution generation

```

1: Input:  $w_j$ 's,  $W$ ,  $E$ ,  $|\mathbb{T}|$ ,  $\sigma$ 
2: Output: eVTOL flight schedule  $F$  and package-flight assignment  $U$ 
3: Calculate  $A_k$  for all  $k \in \mathbb{K}$ 
4: Rank vertiports based on  $A_k$ 's
5: for  $t = 1$  to  $|\mathbb{T}|$  do
6:    $|F|^t \leftarrow$  Eq. (32)
7:   Dispatch  $|F|^t$  flights to the top  $|F|^t$  ranked vertiports
8:   while at least one  $j \in \mathbb{J}$  with  $t_j^l \leq t$  has not been examined do
9:     Select one such  $j$  with largest  $\eta_j$ 
10:    while at least one  $f \in F$  with  $t_f \leq t_j^l$  has not been considered do
11:      Select one such  $f = (k_f, t_f)$  with least  $d_{j,k_f}$ 
12:      if  $w_j \leq W - \sum_{j' \in \mathbb{J}_f} w_{j'}$  then
13:         $U_j \leftarrow f$ 
14:      end if
15:    end while
16:  end while
17: end for

```

Algorithm 1 takes as inputs package weights w_j 's, eVTOL carrying capacity W , eVTOL fleet size E , the number of time stamps $|\mathbb{T}|$, and the weight parameter for vertiport ranking σ . The output is an initial eVTOL flight schedule and an initial assignment of packages to flights. The algorithm starts by calculating the attractiveness of all vertiports for a given σ value using Eq. (31) (lines 3) and rank the vertiports (line 4). Then, for each time stamp, the number of flights to dispatch $|F|^t$ is estimated using (32) (line 5–6). For $t = 1$, the number of flights to dispatch is the minimum of the ideal number of needed flights $\lceil \sum_{j \in \mathbb{J}, t_j^l = 1} w_j / W \rceil$ and the fleet size E . For $t \geq 2$, the number of flights to dispatch is the sum of a deterministic component $|F|_1^t$ and a random component $|F|_2^t$. $|F|_1^t$ is an estimate of the number of needed flights based on the packages that must be assigned by the time stamp and the

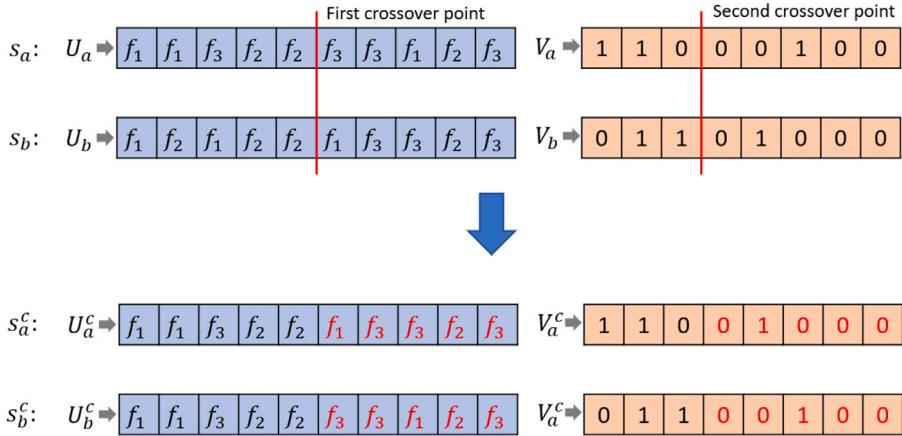


Fig. 5. Illustration of crossover performed on two solutions s_a and s_b , to generate two new solutions s_a^c and s_b^c .

number of available eVTOLs. $|F|_2^t$ introduces randomness to diversify the solutions by accounting for the noise impact. Details about determining the two components are presented in [Appendix B](#).

After determining $|F|^t$, these flights are dispatched towards the top $|F|^t$ ranked vertiport (line 7), given that a vertiport can accommodate at most one flight for a time stamp. In line 8–16, unassigned packages are assigned to the flights in a greedy fashion. Specifically, we examine unassigned packages j with $t_j^l \leq t$ one by one, in decreasing order of latest delivery time η_j (line 8–9). For an unassigned package j , we check if there is a flight dispatched no later than t_j^l (line 10) towards the vertiport closest to j 's destination (line 11) and has available capacity to carry j (line 12). \mathbb{J}_f in line 12 denotes the set of packages already assigned to flight f . If the check says yes, assign package j to the flight (line 13) (in the case that multiple such flights exist, pick the earliest dispatched flight). If the check says no, look for flights dispatched towards the second closest vertiport (i.e., proceed to the next iteration in the “while” loop of line 10), and so on until a flight is found. In the rare case that no flight can be found, package j is left unassigned for the next time stamp.

$$|F|^t = \begin{cases} \min(\lceil \sum_{j \in \mathbb{J}, t'_j=1} w_j / W \rceil, E) & \text{if } t = 1 \\ |F|_1^t + |F|_2^t & \text{if } t \geq 2 \end{cases} \quad (32)$$

Note that the above process of generating an initial solution involves randomness (estimating the flight number for $t \geq 2$) and how vertiports are ranked, the latter depending on the value of parameter σ . In view of these, we repeat the algorithm a number of times, to generate multiple initial solutions under a given σ value. In our numerical experiments, 20 σ values are taken from 0 to 1 with an equal increment. Recalling that in total $2G$ initial solutions are needed, under each σ value the algorithm is performed $G/10$ times. In case a solution reappears, we apply a random mutation described in [Section 5.5](#) to create a different solution, so that the total number of generated solutions is $2G$.

5.3. NSGA2 operations

Given the population of solutions at the beginning of iteration i , i.e., \mathbb{S}_i in [Fig. 3](#), we follow the classic NSGA2 algorithm to perform three operations in sequence: (1) non-dominated sorting and front generation, (2) crowding distance calculation, and (3) preserving elite solutions ([Deb et al., 2002](#)), resulting in an elite population of solutions. The elite population is denoted by \mathbb{S}'_i . As these operations are standard, they are only briefly described in [Appendix C](#).

5.4. Local search

To improve the quality of solutions in the elite population, three customized local search heuristics: (1) add a flight, (2) move a flight, and (3) remove a flight are designed and performed on each solution in the elite population.

5.4.1. Add a flight

The objective of adding a new flight is to try to reduce shipping cost by moving some packages to a newly added flight, to reduce distance violation and consequently total shipping cost. Details of the heuristic is presented in [Algorithm 2](#). Line 3–9 selects the time stamp and the vertiport to dispatch a new flight f , based on population presence (ρ_t) and vertiport attractiveness (A_k). With f , we compute the distance violation for each package j if moving j from its currently assigned flight to f , which is $v'_j = \max\{0, d_{j,k_f} - \beta d_j^*\}$ (line 10). Δv_j , the distance violation reduction for j compared to its current distance violation (which is $v_j = \max\{0, d_{j,k_{U_j}} - \beta d_j^*\}$) is

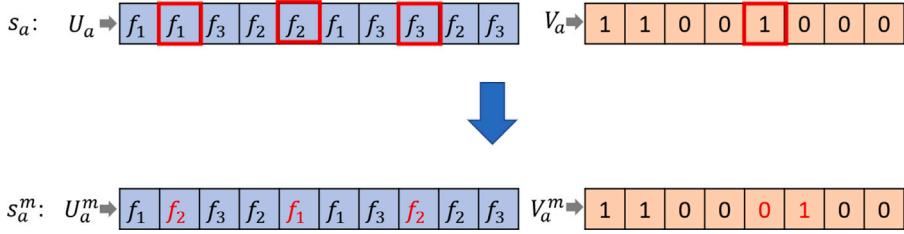


Fig. 6. Illustration of one solution mutation.

subsequently computed (line 11). Then, we explore moving packages to flight f one by one, in decreasing order of Δv_j (line 12–17). A package j is moved to f only when three conditions are met: (1) distance violation is reduced; (2) the latest time stamp by which j leaves the warehouse is no later than t ; and (3) f has available capacity (line 14). If these conditions are met, j is moved to f (line 15). If this leads to a reduced cost, we update the solution (line 19). If not, we keep the current solution (line 21). Note that after a flight is added, the noise impact is always increased. This is acceptable as the updated solution may still be on the Pareto frontier, especially if the noise impact increase is small. If a flight is added, the updated solution is added to the elite population.

Algorithm 2 “Add a flight” heuristic

```

1: Input: current solution
2: Output: updated solution
3: while no flight is added and at least one  $t \in \mathbb{T}$  has not been examined do
4:   Select  $t$  that has not been examined with least  $\rho_t = \sum_{k \in \mathbb{K}} \rho_k^t$ 
5:   Select vertiport  $k \in \mathbb{K}$  with least  $A_k$  and available for flight dispatching at  $t$ 
6:   if an eVTOL is available for dispatching at  $t$  then
7:     Add a new flight  $f = (k_f, t_f)$  with  $k_f = k$  and  $t_f = t$ 
8:   end if
9: end while
10: Compute  $v'_j, \forall j \in \mathbb{J}$ 
11: Compute  $\Delta v_j = v_j - v'_j, \forall j \in \mathbb{J}$ 
12: while at least one  $j \in \mathbb{J}$  has not been examined do
13:   Select one such  $j$  with largest  $\Delta v_j$ 
14:   if  $\Delta v_j > 0$  and  $t_j^l \leq t_f$  and  $w_j \leq W - \sum_{j' \in \mathbb{J}_f} w_{j'}$  then
15:      $U_j \leftarrow f$ 
16:   end if
17: end while
18: if shipping cost is reduced then
19:   Update current solution by the new solution
20: else
21:   Do not update
22: end if

```

5.4.2. Move a flight

In this heuristic, we move the destination of an existing eVTOL flight to a different vertiport. The hope is to improve both cost and noise objectives. Details of the heuristic is given in Algorithm 3. In performing the algorithm, we first pick the flight with the largest distance violation (v_f in line 4, which is calculated as $\sum_{j \in \mathbb{J}_f} v_j$). Fixing the dispatching time stamp of the flight, we look for another vertiport that is available, does not increase the noise impact (line 5), and leads to the most cost reduction if the flight is moved to the alternative vertiport (line 7). Note that as the flight dispatching time stamp is fixed, there is no change in package time violation. If such a move with a positive cost reduction can be found, it will be a Pareto improvement move. Perform the move and update the solution (line 8–11). Otherwise, no move is performed (line 11–12). If a flight can be moved, the updated solution replaces the current solution in the elite population.

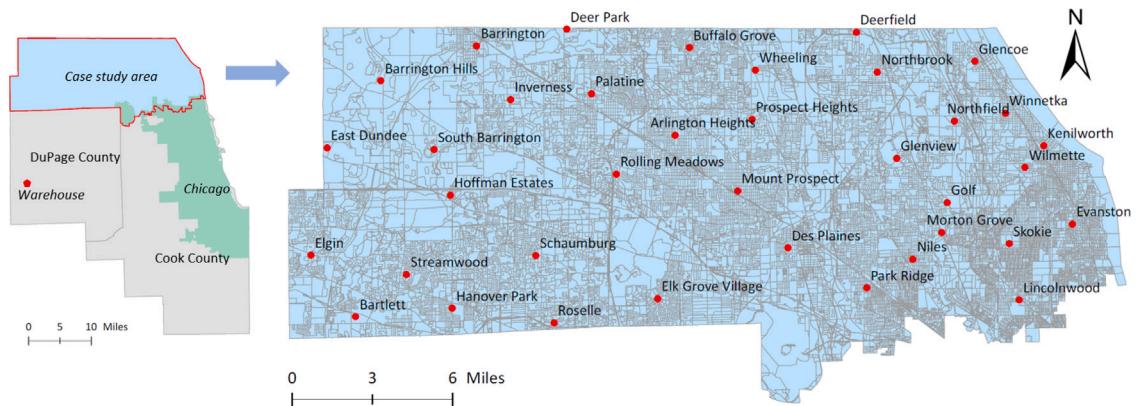


Fig. 7. Illustration of the study area.

Algorithm 3 "Move a flight" heuristic

```

1: Input: current solution
2: Output: updated solution
3: while no flight is moved and at least one flight  $f \in F$  has not been considered do
4:   Select one such  $f = (k_f, t_f)$  with largest  $v_f$ 
5:   Identify the set of available vertiports  $\{k'\}$  with  $N_{k'}^t \leq N_{k_f}^t$ 
6:   if  $\{k'\} \neq \emptyset$  then
7:     Select  $k'^* \in \{k'\}$  that reduces shipping cost most by letting  $f = (k'^*, t_f)$ 
8:     if cost reduction is positive then
9:        $f \leftarrow (k'^*, t_f)$ 
10:      Update current solution
11:    else
12:      Do not update
13:    end if
14:  end if
15: end while

```

5.4.3. Remove a flight

The intent of this heuristic is to reduce eVTOL flying cost by removing one eVTOL flight. All packages carried on the removed flight need to re-assigned to other flights. Details of the heuristic is given in Algorithm 4. Specifically, the heuristic picks the flight with the least carried weight to remove (line 3), as accommodating packages on this flight by the remaining flights would likely be easier. If the remaining flights have sufficient available capacity to accommodate these packages (line 4), check the packages one by one in decreasing order of the latest delivery time to see if it is possible to re-assign a package to another flight (line 5–6). The check and re-assignment (line 7–12) take a similar procedure as that of line 10–15 in Algorithm 1. If all packages originally on the picked flight are re-assigned, update the solution (line 17–18). Otherwise, we keep the current solution (line 19–20 and 22–23). If a flight can be removed, the updated solution replaces the current solution in the elite population.

After performing the three local search heuristics on all elite solutions in \mathbb{S}'_i , the elite population becomes \mathbb{S}''_i . The size of \mathbb{S}''_i is between G and $2G$. The lower bound G occurs to the extreme case that no new solution is added after performing “adding a flight” heuristic. The upper bound $2G$ occurs to the other extreme case that performing the “adding a flight” heuristic always adds a new solution for each elite solution in \mathbb{S}'_i . With \mathbb{S}''_i , we proceed to crossover and mutation operations to evolve the population by generating offspring solutions.

Algorithm 4 “Remove a flight” heuristic

```

1: Input: current solution
2: Output: updated solution
3: Select flight  $f \in F$  with least carrying weight
4: if  $(|F| - 1)W - \sum_{j \in \mathbb{J}_f} w_j \geq \sum_{j \in \mathbb{J}_f} w_j$  then
5:   while at least one  $j \in \mathbb{J}_f$  has not been examined do
6:     Select one such  $j$  with largest  $\eta_j$ 
7:     while at least one  $f' \in F \setminus \{f\}$  with  $t_{f'} \leq t_j^l$  has not been considered do
8:       Select one such  $f' = (k_{f'}, t_{f'})$  with least  $d_{j, k_{f'}}$ 
9:       if  $w_j \leq W - \sum_{j' \in \mathbb{J}_{f'}} w_{j'}$  then
10:         $U_j \leftarrow f'$ 
11:      end if
12:    end while
13:    if no flight can be found then
14:      Break
15:    end if
16:  end while
17:  if all packages originally in  $\mathbb{J}_f$  are re-assigned then
18:    Update current solution
19:  else
20:    Do not update
21:  end if
22: else
23:  Do not update
24: end if

```

5.5. Offspring generation

To generate offspring solutions, we iteratively select pairs of solutions from \mathbb{S}_i'' to perform crossover-mutation operations. More specifically, at each iteration we randomly pick two solutions from \mathbb{S}_i'' , on which a tournament is applied to select one as s_a . Similarly, we select another solution as s_b . With s_a and s_b , a probability is applied to determine whether crossover is performed. If yes, we randomly pick a crossover point in each of the two binary vectors for s_a and s_b , so that after the crossover point the elements in the two solutions are swapped. Fig. 5 gives an illustrative example, where the red parts are the swapped elements. This results in two new solutions s_a^c and s_b^c . We then check the feasibility of s_a^c and s_b^c , in terms of eVTOL availability (the number of eVTOL flights dispatched at a time stamp no greater than the available eVTOLs) and carrying capacity constraints. If both s_a^c and s_b^c are feasible, we continue to perform mutation on s_a^c and s_b^c . If at least one of s_a^c and s_b^c is infeasible, mutation is performed on s_a and s_b . In either case, the mutated solutions are added to \mathbb{S}_i'' . Below we describe how mutation is performed.

Given a solution $s_a = (U_a, V_a)$, mutation proceeds as follows. For U_a , we first randomly pick a pre-specified number of packages. As each package is associated with a flight to which the package is assigned, for each picked package we randomly choose another flight and re-assign the package to the flight. Let us use an example to illustrate this (Fig. 6). This example considers 10 packages, two vertiports, and four time stamps (so U_a has 10 elements; V_a has $2 \times 4 = 8$ elements). Three packages: 2, 5, and 8, are randomly picked. In U_a , package 2 is assigned to flight f_1 , package 5 is assigned to f_2 , and package 8 is assigned to f_3 . Now, we randomly choose another flight for package 2, which is f_2 in Fig. 6. Similarly, package 5 is re-assigned to f_1 , and package 8 is re-assigned to f_2 . The new U vector becomes U_a^m . After each re-assignment, we always update the load of each flight and check whether the eVTOL carrying capacity is exceeded. If an re-assignment leads to capacity violation, the re-assignment is not performed. In the rare case that all the intended re-assignments in a mutation leads to capacity violation, the mutation is not performed. We keep U_a as it is.

For V_a , mutation is performed by randomly picking one flight (i.e., one of the “1” elements) and move the flight to a “0” position that corresponds to the same vertiport but a different time stamp (i.e., exchange the “1” with the “0”), while respecting the eVTOL fleet availability constraint. We again use Fig. 6 to illustrate. There are only three “1”s in V_a : per the solution representation as described in Section 5.1, flights f_1 and f_2 are dispatched towards vertiports 1 at time stamps $t = 1$ and $t = 2$ respectively, while flight f_3 is dispatched towards vertiport 2 at time stamp $t = 1$. In the example, the randomly picked flight is f_3 . f_3 is moved to time stamp $t = 2$ for the same vertiport, which is the sixth position in V_a . The resulting new vector is V_a^m . In the rare case that no other time stamp can accommodate the picked flight, mutation is not performed on V_a . We keep V_a as it is.

Each time we perform the above crossover-mutation operations, typically two new solutions will be added to \mathbb{S}_i'' (there can be cases that infeasibility in one or both of the mutated solutions results in only one or no new solutions added). As we iteratively perform the above crossover-mutation operations, we will reach a point that the size of \mathbb{S}_i'' becomes either $2G$ or $2G - 1$. In the latter case, we randomly pick a solution from \mathbb{S}_i'' and perform only mutation to generate a new feasible solution, which is then added to \mathbb{S}_i'' . In the end, the size of \mathbb{S}_i'' after the crossover-mutation operations is $2G$.

After the above crossover-mutation operations, we perform additional mutations to further enhance the diversity of solutions. Specifically, m_i non-Pareto solutions are randomly picked from the current population, where m_i is equal to the total number of flights over all solutions in the population multiplied by a mutation rate $\mu \in (0, 1)$. Here we consider the number of flights because, as described above, each mutation involves changing the dispatching time stamp of only one flight of one solution. We focus on mutating non-Pareto solutions as our numerical experiments suggest that mutation of the Pareto solutions has little effect on improving the Pareto frontier. For each picked solution, the above mutation procedure is performed. If a mutated solution is feasible, it replaces the picked solution. After these additional mutations, the resulting population becomes \mathbb{S}_i''' .

In the next section, we implement the bi-objective optimization model with the NSGA2-GG-LS algorithm in a case study of package delivery in the north suburbs of the Chicago metro area. In addition to this section, the performance of the NSGA2-GG-LS algorithm is compared with alternative solution methods, including (1) variants of NSGA2-GG-LS, (2) NSGA3-GG-LS in which NSGA3 is an extension of NSGA2, (3) the ϵ -constrained method, and (4) a semi-exhaustive enumeration method. Details about the algorithmic comparison are presented in Appendices D–F.

6. Numerical experiments

The numerical experiments are conducted in the context of package delivery from a warehouse to customers in the north suburb of the Chicago metro area (Fig. 7). The study area has 37 census places. We consider that each census place has a vertiport located at its center, displayed by red dots in the right-side map of Fig. 7. According to the National Household Travel Survey (US Department of Transportation, 2017), the study area has over 440,000 households and on average 70,000 packages to deliver from online shopping each day. In view of these statistics and the fact that typically multiple delivery service providers serve an area, we solve a range of different problem sizes up to 10,000 package deliveries in a day.

Specifically, three demand levels are examined: 1000, 5000, and 10,000 packages. Corresponding to the three demand levels, we consider the eVTOL fleet size to be 5, 20, and 25 eVTOLs. The eVTOL carrying capacity is 500 lbs, which is in the range of carrying capacities of the existing cargo-carrying eVTOL designs in Table 1. The destination of a package is from a randomly picked household which, for numerical convenience, is assumed to be located at the center of the census block in which a household is located. The weight of each package is randomly drawn from the range of 1–10 lbs. The latest delivery time of a package (η_j) is also randomly generated between 10 am and 10 pm. The numerical experiments are performed on a PC with the 11th Generation Intel(R) Core(TM) i5-1135G7, 2.40 GHz processor, 8.00 GB of RAM, and 64-bit Windows 11 operating system.

6.1. Parameter values

We choose eVTOL-related parameter values based on the information in the literature whenever possible. In terms of eVTOL noise characterization, we follow Archer Aviation (2021) and consider a sound level of 45 dB on the ground when an eVTOL is vertically 2000 ft above. For vertical ascent/decent, eVTOL speed is assumed 200 ft/minute as suggested in Antcliff et al. (2016). The safe and comfortable vertical acceleration and deceleration of eVTOL is considered 20% of gravitational acceleration (Beyne and Castro, 2022). With these values, the distance and time required for an eVTOL to accelerate from 0 to 200 ft/minute (or decelerate from 200 ft/minute to 0) would be very small (less than one foot and about half a second). Thus, for simplicity an eVTOL is considered to have a constant speed of 200 ft/minute during vertical ascent and descent.

In terms of eVTOL operations other than noise generation, we consider that eVTOL dispatching is performed periodically every two hours in a day starting from 8 am till 6 pm. In other words, $\Delta = 2$ h and six time stamps are considered. eVTOL turnaround time at a vertiport is set to be 10 min: $\phi_1 = 10$ min. The time for eVTOL check/charging and package loading at the warehouse is 20 min: $\phi_2 + \phi_3 = 20$ min. For parameters β and α which are related to maximum distance and latest time stamp constraints, we set them as $\beta = 3$ and $\alpha = 2$ h.

The operating cost of an eVTOL flight consists of eVTOL flying cost and vertiport usage fee. The eVTOL flying cost is computed by summing the operating cost of individual phases of a flight, given that different phases are associated with different power requirement. Specifically, the average power requirement during the vertical ascent and descent phases is calculated as $P_1 = \frac{mg}{\eta_v} \sqrt{\frac{\delta}{2\rho}}$, where m is the eVTOL aircraft mass, g is the gravitational acceleration constant, η_v is system efficiency during vertical flight conditions, δ is disk loading, and ρ is sea-level air density.³ During the cruise phase, the required power is calculated as $P_2 = \frac{mgV}{L/D \eta_c}$, where L/D is lift-to-drag ratio, V is cruise speed, and η_c is system efficiency during cruise flight conditions. The ratio of the required

³ Following momentum theory and standard equations of motion, the power requirement in vertical flight for open rotor aircraft (which is assumed in the paper) is $\left[\frac{f mg}{FoM} \sqrt{\frac{A}{2\rho}} + \frac{mgV_{climb}}{2} \right] / \eta_{mech}$ (Johnson, 1994; Sripad and Viswanathan, 2021; Sripad, 2022). In this expression, f is a correction factor for interference from the fuselage, FoM is the figure of merit which is the ratio between ideal and actual rotor power, A is the disk area, V_{climb} is the rate of climb, and η_{mech} is the electromechanical efficiency of the motors and electric powertrain during vertical flight conditions. Given that V_{climb} is positive for ascent and negative for descent and under the assumption of symmetric operations, the V_{climb} -related terms cancel out between vertical ascent and descent. Thus, the average power

powers during the two phases is therefore $\frac{P_1}{P_2} = \frac{1}{V} \frac{L}{D} \frac{\eta_c}{\eta_h} \sqrt{\left(\frac{\delta}{2\rho}\right)}$. We assume an eVTOL cruise speed of 100 mph (44.7 m/s). Drawing the other parameter values from Kasliwal et al. (2019): $L/D = 17$, $\eta_c = 0.765$, $\eta_v = 0.63$, $\delta = 450 \text{ N/m}^2$, and $\rho = 1.225 \text{ kg/m}^3$, the ratio is 6.259.

We follow Joby Aviation (2021) and consider an average operating cost of \$2/mile during cruise. With a cruise speed of 100 mph, the time-based unit operating cost is \$3.33/minute. Assuming that eVTOL operating cost per unit time is proportional to the power required, the time-based unit operating cost during ascent/descent is \$20.84/minute (6.259 times \$3.33/minute). We consider cruise altitude of 1500 ft. Using the constant ascent/descent speed of 200 ft/minute, vertical ascent/descent each take 7.5 min, amounting to \$312.6 per ascent and descent. For the inclined ascent and inclined descent phases, they are assumed to take 2 min (Kasliwal et al., 2019) with the same unit operating cost as for cruise. The usage fee per eVTOL operation at a vertiport is randomly drawn from the range of \$10-30, to reflect the possible price heterogeneity among the different vertiports. The two penalty parameters in the objective O_1 are set as $\theta_1 = \$0.5/\text{mile}$ and $\theta_2 = \$2/\text{time stamp}$.

In implementing NSGA2-GG-LS, a few parameters need to be specified. The number of solutions in a population is set to 100 (i.e., $G = 50$). The maximum number of iterations is set to 10. This number might seem small at first sight. However, recall that we start with some good initial solutions through guided population generation. Moreover, at each iteration solutions in the elite population are improved by local search heuristics. Because of these, we find that the search for the Pareto frontier only needs a few iterations. In performing “move a flight” heuristic, the number of flights to be tried is set to three. In performing crossover, the tournament probability is set to 0.9. The crossover probability is set to 0.5. For mutation, the number of packages to pick is five. The mutation rate μ is set to 0.2 which is shown to generate the best Pareto frontier results after testing different values for μ in $(0, 1)$.

6.2. Community noise impact estimates for a single eVTOL operation

With the sound level at 45 dB on the ground when vertically under an eVTOL at a height of 2000 ft, we first compute the threshold height H_n (see Fig. 2) that $L_{H_n}(0) = L_a$, which is 50 dB. Using Eq. (5), we obtain $H_n = 1,125 \text{ ft}$, which is much lower than the cruise altitudes of many eVTOLs reported in the literature, ranging between 1476 to 9842 ft (Sripad and Viswanathan, 2021; Jeong et al., 2021). We note that H_n is also way below the maximum cruise altitudes of the eVTOL types in Table 1. For example, 3000 ft for Beta Alia; 22,000 ft for Sabrewing Rhaegal (this is cruise altitude); 9843 ft for Starling Cargo, and 19,700 ft for Pipistrel Nuova V300 (Guisbond, 2021; Sabrewing Aircraft Company, 2021; Vertical Flight Society, 2022; Pipistrel, 2022). The number of height intervals is set to 10, i.e., $n = 10$. Per the description in Section 3.2, the radius of the impacted area surrounding the vertiport where an eVTOL operation occurs is also equal to H_n . The size of the impacted area is thus $3.14 \times 1125^2 = 3.97 \times 10^6 \text{ sq. ft}$, or 0.14 sq. miles. Given the constant speed of 200 ft/minute during vertical ascent/descent, the time an eVTOL spends in each height interval is the same, at $T_j = 1125/10/200 = 0.56 \text{ min. } \forall j = 1, \dots, 10$.

We use census block-level data to estimate the population in the impacted area surrounding each vertiport. For ρ_k^i 's, given that the precise values are not available to us, we instead use the following assumed values: 1.0 for 8–10 am, 0.6 for 10 am–12 pm, 0.4 for 12–4 pm, 0.8 for 4–6 pm, and 1.0 after 6 pm for all vertiports, to reflect that some people leave home for work in the morning and return home in the evening. Applying Eq. (13) yields the population exposure to noise from one eVTOL operation at each vertiport when the eVTOL flight is dispatched at a time stamp. Note that given the eVTOL speed and the distance between the warehouse and the vertiports, an eVTOL dispatched at a time stamp will always land at a vertiport within the next two hours.

6.3. Optimization results

As package demand is randomly generated, multiple problem instances (PIs) are solved. Specifically, nine PIs of three problem sizes are solved: three PIs with 1000 packages and five eVTOLs, three PIs with 5000 packages and 20 eVTOLs, and three PIs with 10,000 packages and 25 eVTOLs. Fig. 8 presents the Pareto frontiers. A common feature of these frontiers is that the variations of the community noise impact is much greater than the variations of shipping cost. For example, for PI 4, the largest shipping cost value is 1.2 times the smallest shipping cost value on the frontier (about \$74000 vs. \$62,000). In contrast, the largest community noise impact is 13.6 times the smallest community noise impact (about 95,000 person-minutes vs. 7000 person-minutes in population exposure). Thus, it may be possible to significantly reduce the noise impact with a relatively small change in the shipping cost.

Looking more closely, we observe well-defined elbow points on the frontiers across the PIs for the demand sizes of 1000 and 5000. The elbow points indicate a prominent pattern change in terms of the variation of one objective value relative to the other. On the left of an elbow point, the steep slope suggests that a small increase in the shipping cost will result in a relatively large decrease in the noise impact. This is not surprising because when shipping cost is low, the emphasis is on making shipping cheap with little regard to the eVTOL noise impact. As a result, the noise impact is very high. At this stage, even paying slightly more attention to noise can reduce the noise impact significantly. On the right side of an elbow point, the flatter slope suggests that with further increase in the shipping cost, the noise impact reduction is marginal. This can also be intuitively explained: when the eVTOL noise impact on communities is already low, further reducing noise would be difficult. For the demand size of 10,000, the presence of an

requirement during vertical ascent and descent is $\left[\frac{fmg}{\text{FoM}} \sqrt{\frac{fmg}{2\rho}} \right] / \eta_{\text{mech}}$. Further defining $\eta_v = \frac{\text{FoM}}{f\sqrt{f}} \eta_{\text{mech}}$ and expressing disk loading mg/A by δ , the average power requirement expression becomes $\frac{mg}{\eta_v} \sqrt{\frac{\delta}{2\rho}}$.

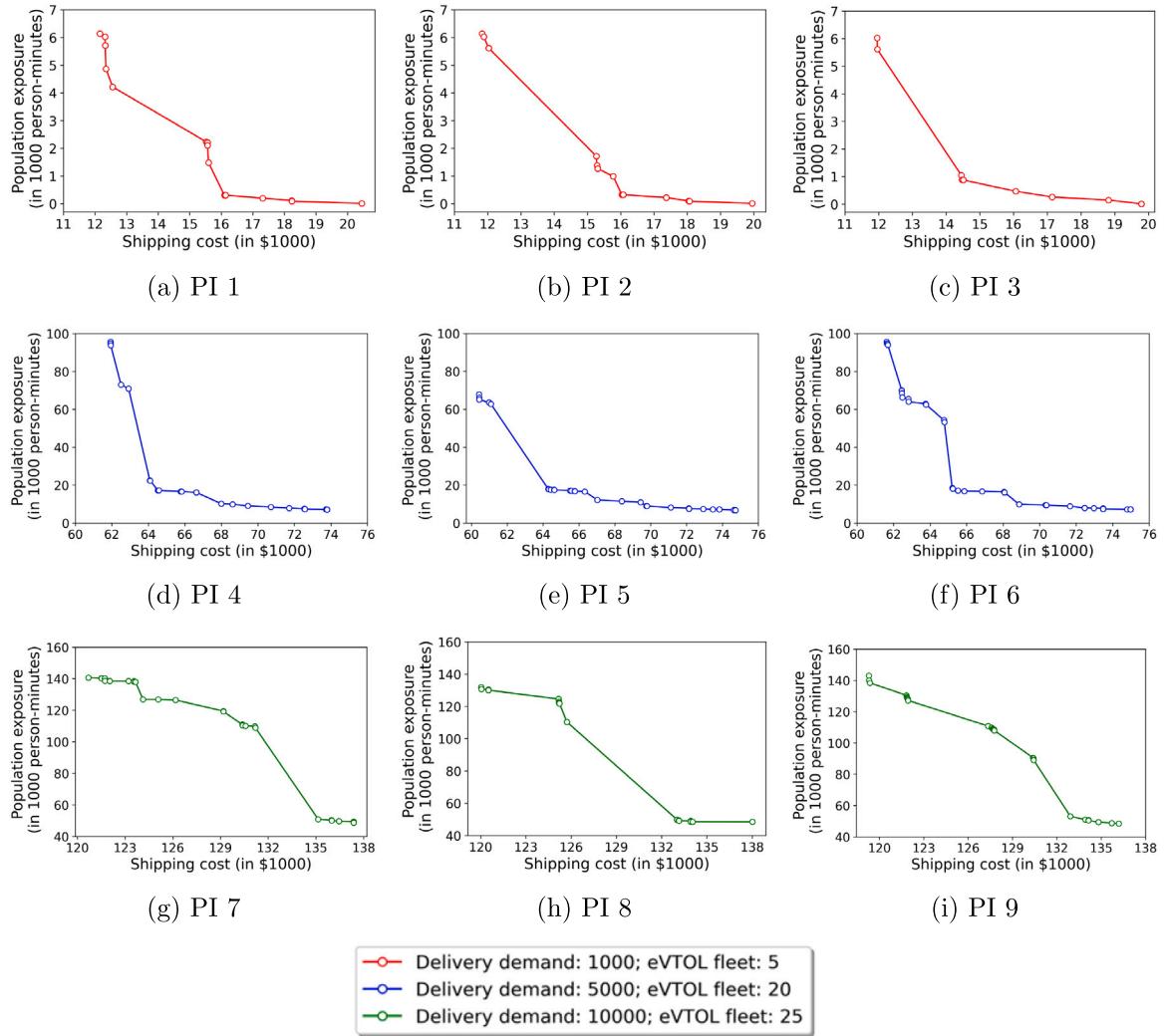


Fig. 8. Pareto frontiers from solving nine problem instances of three sizes.

elbow point is less obvious than other two demand sizes. An explanation is that with a larger problem size, the solution space also gets larger leading to more possibilities of simultaneously varying shipping cost and noise impact, making their changes smoother and more continuous.

To further investigate the trade-off between shipping cost and community noise impact, the following experiment is conducted. For each of the nine PIs, we start from the Pareto frontier solution with the largest noise impact, and examine the cost reduction when decreasing the noise impact by different values on the frontier. Given the different population exposure for the three problem sizes, the decrease in noise impact is considered differently. For PI 1-3, we consider decreasing the noise impact value by 1000, 2000, and 3000. For PI 4-6, the decreases are 10,000, 30,000, and 60,000. For PI 7-9, the decreases are 10,000, 50,000, and 90,000. Table 2 shows the results. In the table, the percent decrease in community noise impact with respect to the largest noise impact is displayed in the third column. The percent increase in shipping cost vis-à-vis the smallest shipping cost (which is associated with the largest noise impact) on the frontier and the resulting cost increase per package are also computed, shown in the last two columns. The cost increase per package is interesting as it could be passed onto customers in the form of higher shipping fees. Alternatively, the cost increase could be viewed as the needed government subsidy to reduce the community noise impact without letting customers bear the economic consequence of doing so.

We find that, for PIs with 1000 packages, 1000 population-minutes of exposure only account for about 16% of the original noise impact. With such a reduction, the increase in shipping cost is minimum, by at most 5% and 59 cents per package. Reducing the noise impact by 2000 person-minutes exposure, or about 33% of the original level, would incur 4%-12% cost increase, which translates to an average cost increase of 94 cents per package. Further reducing the noise impact by 3000 person-minutes exposure, or about 50% of the original level, would cost each package on average \$1.89 more across the three PIs. Similar trends can be said

Table 2

Trade-off between community noise impact and shipping cost increase.

| Problem instance | Decrease in community noise impact (exposure) | % decrease in community noise impacts | % shipping cost increase | Cost increase per package (\$) |
|------------------|---|---------------------------------------|--------------------------|--------------------------------|
| PI1 | 1000 | 16.3 | 1.5 | 0.18 |
| | 2000 | 32.6 | 4.3 | 0.52 |
| | 3000 | 48.9 | 16.6 | 2.01 |
| PI2 | 1,000 | 16.3 | 5.0 | 0.59 |
| | 2000 | 32.6 | 12.0 | 1.42 |
| | 3000 | 48.9 | 19.1 | 2.25 |
| PI3 | 1,000 | 16.6 | 2.7 | 0.32 |
| | 2000 | 33.2 | 7.3 | 0.87 |
| | 3000 | 49.8 | 11.8 | 1.41 |
| PI4 | 10,000 | 12.8 | 0.7 | 0.09 |
| | 30,000 | 38.4 | 2.2 | 0.27 |
| | 60,000 | 76.7 | 4.4 | 0.55 |
| PI5 | 10,000 | 10.5 | 0.4 | 0.05 |
| | 30,000 | 31.4 | 1.8 | 0.22 |
| | 60,000 | 62.7 | 3.0 | 0.37 |
| PI6 | 10,000 | 14.7 | 1.7 | 0.20 |
| | 30,000 | 44.2 | 4.0 | 0.49 |
| | 60,000 | 88.4 | 18.8 | 2.27 |
| PI7 | 10,000 | 7.1 | 2.7 | 0.33 |
| | 50,000 | 35.5 | 9.7 | 1.17 |
| | 90,000 | 64.0 | 12.2 | 1.48 |
| PI8 | 10,000 | 7.6 | 4.3 | 0.52 |
| | 50,000 | 37.9 | 7.6 | 0.91 |
| | 90,000 | 60.6 | 10.6 | 1.27 |
| PI9 | 10,000 | 7.0 | 1.5 | 0.18 |
| | 50,000 | 35.0 | 9.0 | 1.07 |
| | 90,000 | 62.9 | 11.5 | 1.37 |

about the other two problem sizes. Overall, starting from a solution with the most community noise impact, we find that the percent reduction in noise impact would be far greater than the percent increase in shipping cost. With the greatest reduction considered for community noise impact, the shipping cost increase would be at most \$2.27 per package from the nine PIs investigated.

In addition to the trade-off experiment, a comparison of eVTOL operations is performed between the two extreme solutions on each Pareto frontier. Fig. 9 shows the number of flights dispatched at each time stamp for the nine PIs. The blue bars correspond to the cost-minimum solutions, while the orange bars are associated with the noise impact-minimum solutions. We observe that flight dispatching tends to be more concentrated towards the middle of a day for cost-minimum solutions. As opposed to this, the flights dispatched are more evenly distributed in a day for noise impact-minimum solutions. Such distinct dispatching patterns may be interpreted by the fact that inter-veriport noise impact difference is much greater than inter-temporal difference while sticking to one veriport. Thus, when noise impact minimization is the focus, a more natural choice is to fly to those low noise impact veriports, rather than to high noise impact veriports at time stamps of lower noise impact. Considering further that at one time stamp at most one flight is dispatched towards a veriport, the strategy of dispatching flights to the low noise impact veriports will be spread out using many time stamps. On the other hand, if cost minimization is the focus and a flight has the flexibility to be dispatched at multiple time stamps while incurring the same low cost, the time stamp with lower noise impact will be chosen. This explains an overall trend of more flight dispatching towards the middle of a day, when the noise impact is lower due to a smaller fraction of population (lower ρ_k') in the impacted communities.

The load factors of eVTOL flights for the two extreme solutions on each Pareto frontier are also compared (Fig. 10). For a flight, the load factor is calculated as the weight carried divided by the eVTOL carrying capacity. In Fig. 10, each bar gives the average load factor over all dispatched flights for a solution. We observe that the average load factor does not vary significantly by problem instance, nor by priority of the optimization objectives. All load factor values are very close to 100%. The results suggest that maximizing utilization of eVTOL carrying capacity is emphasized when dispatching flights. This is easy to understand: otherwise, more flights would be needed leading to greater cost and noise impact.

6.4. Sensitivity analysis

In this section, we examine sensitivity of the Pareto frontier to three key parameters of the model: (1) eVTOL fleet size, (2) eVTOL carrying capacity, and (3) β , which is the multiplier of the maximum distance between the package destination and the veriport a package is sent to, with respect to the minimum distance between the package destination and its closest veriport. The sensitivity analysis is performed on three randomly generated PIs with 1000 package delivery demands and a fleet size of five eVTOLs. The resulting Pareto frontiers are presented in Figs. 11–13.

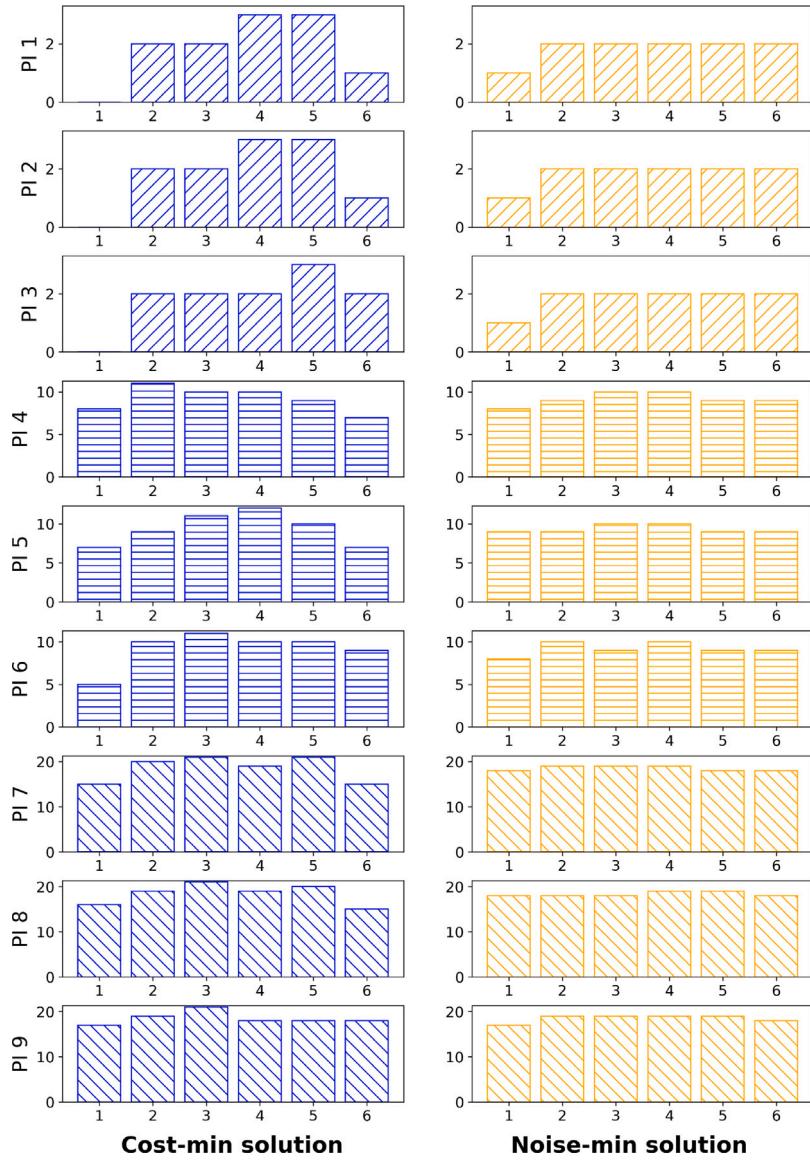


Fig. 9. Number of eVTOL flights dispatched at each time stamp, for the cost-minimum and noise impact-minimum solutions across the nine problem instances.

Fig. 11 presents the sensitivity of the Pareto frontier to fleet size, for which 5, 10, and 15 eVTOLs are tested. As the fleet size increases, the Pareto frontier remains unchanged for all three PIs. In principle, having a larger fleet size means greater total carrying capacity, which can accommodate more packages and helps reduce distance and time penalty cost. However, the sensitivity results suggest that a fleet size of 5 seems sufficient to accommodate the level of demand considered. Also, it is interesting to see that to achieve the lowest community noise impact, the Pareto solutions are almost invariant to the eVTOL fleet size. This is because minimizing noise means using a minimum number of flights, for which having a larger fleet would not help.

Fig. 12 shows the sensitivity of the Pareto frontier to eVTOL carrying capacity, for which 400, 500, and 600 lbs are considered. We find that as eVTOL carrying capacity increases, the part of the Pareto frontier towards the lower end of shipping cost bends down, suggesting an improvement of the frontier. This is intuitive in that with greater carrying capacity, an eVTOL can load more packages in one flight. Consequently, fewer flights need to be dispatched, resulting in both lower shipping cost and community noise impact. Note that here we fix the eVTOL operating cost, to focus on the effect of eVTOL carrying capacity.

On the other hand, the remaining part of the Pareto frontier towards the lower end of the community noise impact appears quite similar under different eVTOL carrying capacities. This is because, as discussed in Section 6.3, when focusing on minimizing noise impact, the strategy is dispatching eVTOL flights towards vertiports with less population in the surrounding area in the course of a day. Those vertiports are often farther away from package destinations than the vertiports that would be chosen if focusing on cost minimization. In addition, the flight dispatching time may violate the latest time by which a package loaded on the flight needs to

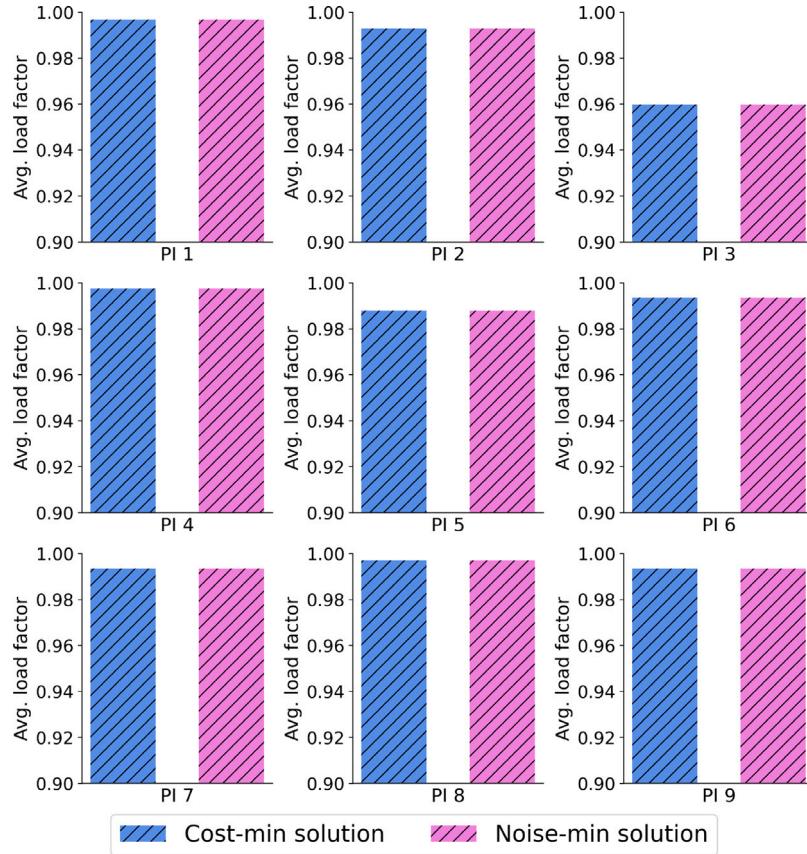


Fig. 10. Average load factor of dispatched eVTOL flights, for the cost-minimum and noise impact-minimum solutions across the nine problem instances.

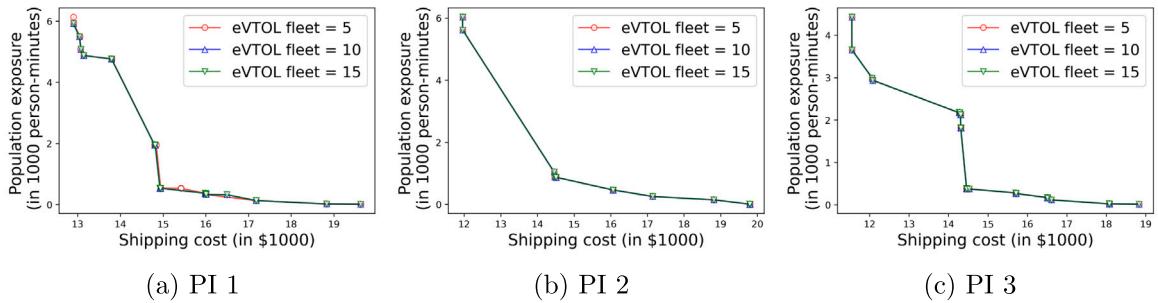


Fig. 11. Sensitivity of Pareto frontiers to eVTOL fleet size.

leave the warehouse. Thus, distance and time penalty costs would incur. In this case, using fewer flights each with a larger carrying capacity to further reduce the noise impact would not be Pareto improving. This helps explain why towards the lower end of the community noise impact, the Pareto frontier does not look very different.

Fig. 13 presents the sensitivity of the Pareto frontier to β , for which four values (1, 2, 3, and 4) are considered. Recall that a smaller β suggests less flexibility for a package to be delivered to a vertiport without incurring distance penalty. Fig. 13 shows that as we increase the value of β , greater flexibility in vertiport choice reduces shipping cost, as manifested by the leftward shift of the Pareto frontier.

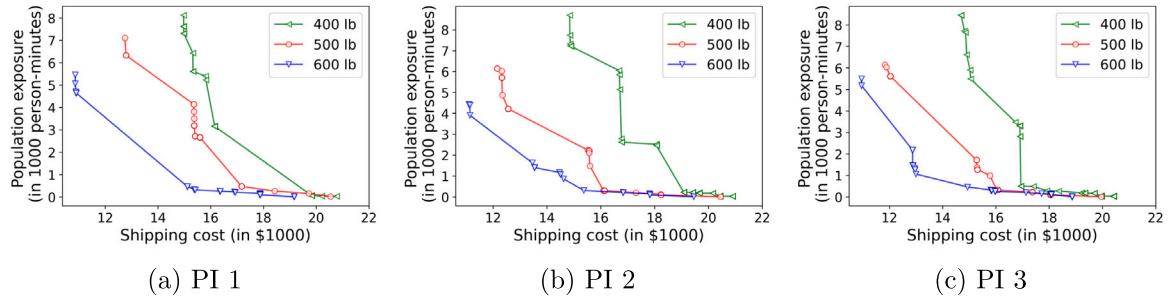
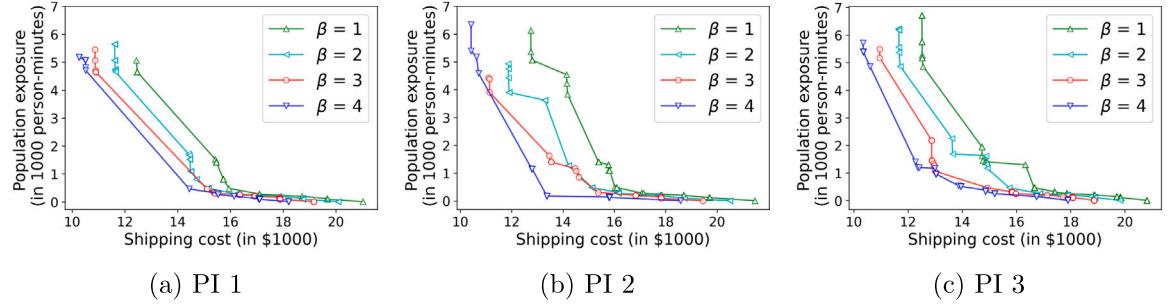


Fig. 12. Sensitivity of Pareto frontiers to eVTOL carrying capacity.

Fig. 13. Sensitivity of Pareto frontiers to β .

7. Conclusion

The rapid development of AAM in recent years suggests a promise to use eVTOLs for package delivery in metro areas. While it is natural to seek economic efficiency in package delivery, the noise impacts of eVTOL operations at vertiports on surrounding communities are important to public acceptance for this new form of delivery. In this paper, we develop a method to quantify the community noise impact of eVTOL operations using a proposed “population exposure” measure, which is based on the sound level generated and accounts for both the number of people impacted and duration of the impact. This measure is incorporated in a bi-objective integer programming model to simultaneously optimize shipping cost and community noise impact of eVTOL operations. A tailored solution algorithm NSGA2-GG-LS, which augments NSGA2 with guided generation of initial population of solutions and local search heuristics, is devised.

The model and algorithm are implemented in a package delivery case study in the Chicago metro area. The results demonstrate the trade-off between minimizing shipping cost and community noise impact, as well as the computational efficiency and effectiveness of the NSGA2-GG-LS algorithm in identifying the trade-off. We find that, starting from the noise-minimum Pareto solution, the percent reduction in community noise impact is much greater than the percent increase in shipping cost as one moves along the Pareto frontier. Along this move, the average cost increase per package is generally small. The numerical results further show that the temporal distribution of eVTOL flight dispatching varies under different system optimization priorities. On the other hand, maximizing utilization of eVTOL carrying capacity is always emphasized. Overall, the proposed method for community noise impact quantification, the bi-objective optimization, and the tailored NSGA2-GG-LS algorithm provide a helpful framework to prepare AAM to be a new form of package delivery that is economically efficient and community friendly.

For extensions of the research, a few directions are suggested. First, as mentioned in Section 3.2, alternative measures to quantify the community noise impact of eVTOL operations could be explored. Second, community surveys could be conducted to obtain more accurate values for ρ_k^t 's. Third, the eVTOL noise impact on individual can vary depending on one is indoor or outdoor. As such, future research could seek to understand (1) the split of outdoor vs. indoor population in communities surrounding each vertiport in the course of a day through community surveys; and (2) the noise impact reduction when people are indoor instead of outdoor, probably by conducting experimental research. Lastly, future research could extend to considering vertiport location choice, especially by taking advantage of the background noise that exist in the surrounding communities. Doing so would require comprehensive, community-level noise maps and the associated data encapsulating all sources of noise. The National Transportation Noise Map in the US and the Environmental Noise Directive in Europe could serve as a starting point for creating such maps.

CRediT authorship contribution statement

Nahid Parvez Farazi: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization. **Bo Zou:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This research was funded by the National Aeronautics and Space Administration (NASA) under Grant Number 80NSSC22K0889. The financial support of NASA is gratefully acknowledged. An earlier version of the paper was presented at the INFORMS 2022 Annual Meeting. We would like to thank the attendees of our presentation as well as the anonymous referees for their constructive comments and suggestions.

Appendix A. Derivation of Eq. (4) and (5)

1. Derivation of Eq. (4):

For two different heights h and h' , we combine the proportional relations (1) and (2) to write the following:

$$\begin{aligned} \frac{P_{h'}^2(0)}{P_h^2(0)} &= \left(\frac{h}{h'}\right)^2 \\ \Rightarrow P_{h'}^2(0) &= \left(\frac{h}{h'}\right)^2 P_h^2(0) \end{aligned} \quad (\text{A.1})$$

Substituting $P_h^2(0)$ in Eq. (A.1) with Eq. (3) yields:

$$P_{h'}^2(0) = \left(\frac{h}{h'}\right)^2 P_o^2 10^{\frac{L_h(0)}{10}} \quad (\text{A.2})$$

Now, we use Eq. (3) to express $L_{h'}^2(0)$ for height h' and then substitute $P_{h'}^2(0)$ with Eq. (A.2), which yields $L_{h'}(0)$:

$$\begin{aligned} L_{h'}(0) &= 10 \lg \frac{P_{h'}^2(0)}{P_o^2} \\ \Rightarrow L_{h'}(0) &= 10 \lg \left(\frac{1}{P_o^2} \left(\frac{h}{h'}\right)^2 P_o^2 10^{\frac{L_h(0)}{10}} \right) \\ \Rightarrow L_{h'}(0) &= 10 \lg \left(\left(\frac{h}{h'}\right)^2 10^{\frac{L_h(0)}{10}} \right) \end{aligned}$$

2. Derivation of Eq. (5):

Below are the steps to derive Eq. (4) from Eq. (5):

$$\begin{aligned} L_{h'}(0) &= 10 \lg \left(\left(\frac{h}{h'}\right)^2 10^{\frac{L_h(0)}{10}} \right) \\ \Rightarrow \frac{L_{h'}(0)}{10} &= \lg \left(\left(\frac{h}{h'}\right)^2 10^{\frac{L_h(0)}{10}} \right) \\ \Rightarrow 10^{\frac{L_{h'}(0)}{10}} &= 10^{\lg \left(\left(\frac{h}{h'}\right)^2 10^{\frac{L_h(0)}{10}} \right)} \\ \Rightarrow 10^{\frac{L_{h'}(0)}{10}} &= \left(\frac{h}{h'}\right)^2 10^{\frac{L_h(0)}{10}} \\ \Rightarrow h'^2 &= h^2 10^{\frac{L_h(0)}{10}} / 10^{\frac{L_{h'}(0)}{10}} \\ \Rightarrow h' &= (h^2 (10^{(L_h(0)-L_{h'}(0))/10}))^{\frac{1}{2}} \end{aligned}$$

Appendix B. Generating $|F|_1^t$ and $|F|_2^t$ for $t \geq 2$ in Algorithm 1

For each of the subsequent time stamps $t = 2, \dots, |\mathbb{T}|$, the number of flights dispatched $|F|^t$ has a deterministic component $|F|_1^t$ and a random component $|F|_2^t$: $|F|^t = |F|_1^t + |F|_2^t$. To compute $|F|_1^t$, we check whether the flights to be dispatched at the previous time stamps still have available capacity. If so, we put as many packages to those flights. If some packages remain, then new flights are added to dispatch at t . The number of newly added flights at t should not exceed the number of available eVTOLs for dispatching at t , which we denote as E^t .

The random component $|F|_2^t$ recognizes that the fraction of people in an area varies in the course of a day. When a smaller fraction of people are in an area at a time stamp (measured $\rho_t = \sum_{k \in \mathbb{K}} \rho_k^t$), the population exposure will also be smaller. Thus, more flights may be considered for dispatching at this time stamp. In view of this, we first normalize ρ_t across all time stamps by min-max scaling, and then generate a random number between 0 and 1 to compare with the normalized ρ_t . If the random number is greater than ρ_t , $|F|_2^t$ additional flights are dispatched at t . $|F|_2^t$ takes a random integer value between 0 and $\min(E^t - |F|_1^t, |F|_{\min} - \sum_{i=1}^{t-1} |F|^i - |F|_1^t)$, where $E^t - |F|_1^t$ gives the number of eVTOL aircraft available at t after dispatching $|F|_1^t$. $|F|_{\min} - \sum_{i=1}^{t-1} |F|^i - |F|_1^t$ is the number of remaining eVTOL flights available for dispatching, which considers (1) the total minimum number of flights $|F|_{\min} = \lceil \sum_{j \in \mathbb{J}} w_j W \rceil$, and (2) the number of flights dispatched thus far ($\sum_{i=1}^{t-1} |F|^i + |F|_1^t$).

Appendix C. NSGA2 operations

Non-dominated sorting and front generation: All the solutions s_i^1, s_i^2, \dots in \mathbb{S}_i are sorted such that a sequence of non-dominated fronts is constructed. Solutions on the first non-dominated front \mathcal{F}_1 constitute the Pareto frontier given the current population \mathbb{S}'_i . Solutions on the second non-dominated front \mathcal{F}_2 constitute a Pareto frontier given \mathbb{S}'_i but excluding solutions on \mathcal{F}_1 . Solutions on the third nondominated front \mathcal{F}_3 constitute a Pareto frontier given \mathbb{S}'_i but excluding solutions on \mathcal{F}_1 and \mathcal{F}_2 , and so on. Following Deb et al. (2002), the non-dominated sorting has a computational complexity of $\mathcal{O}(G^2)$.

Crowding distance calculation: The purpose of calculating crowding distance of each solution is to estimate the density of other solutions surrounding the solution of interest, which helps preserve solutions that are in less crowded region to maintain solution diversity. Specifically, given a solution s , for each objective we first identify s 's two neighboring solutions, i.e., solutions that are on the same front as s and have the nearest objective values larger and smaller than the objective value of s . The difference of the objective values of the two neighboring solutions is divided by the maximum difference of the objective values among all solutions on the front, yielding the crowding distance of the solution with respect to this objective. We then sum the crowding distances over the two objectives, to obtain the crowding distance of the solution. If a solution has an objective value lie at the end of a front, its crowding distance is considered infinite. Following again Deb et al. (2002), the crowding distance calculation has a computational complexity of $\mathcal{O}(G \log G)$.

Preserve elite populations: The results of non-dominated sorting and crowding distance calculation provide the basis for selecting a subset of solutions to form an elite population \mathbb{S}'_i . The size of \mathbb{S}'_i is G , half that of \mathbb{S}_i . Solutions in \mathbb{S}'_i are elite as we select the solutions that come from non-dominated fronts of higher priority. For the last front involved in \mathbb{S}'_i , we select solutions from this front that are in less crowded regions.

Operationally, we first consider solutions on \mathcal{F}_1 as these solutions are not dominated by any other solutions and thus have the highest priority. If the number of solutions on \mathcal{F}_1 is less than G , we proceed to front \mathcal{F}_2 . This continues until we encounter a non-dominated front \mathcal{F}_l such that the number of solutions on $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{l-1}$ is less than G , but the number of solutions on $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_l$ exceeds G . In this case, all solutions on $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{l-1}$ are selected. On \mathcal{F}_l , we select solutions in descending order by crowding distance, such that the number of selected solutions from \mathcal{F}_l plus the solutions on $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{l-1}$ equals G .

Appendix D. Comparison of NSGA2-GG-LS with its variants

In this appendix, we compare NSGA2-GG-LS with its three variants: basic NSGA2, NSGA2 with only guided generation of initial solutions (NSGA2-GG), and NSGA2 with only local search heuristics (NSGA2-LS). The purpose is to understand the impacts of guided generation of initial solutions and local search heuristics, the two salient features of NSGA2-GG-LS, on the quality of the Pareto frontier generated. Fig. D.14 shows the Pareto frontiers for three randomly generated PIs by NSGA2-GG-LS, NSGA2-GG, NSGA2-LS, and NSGA2. These PIs have a demand level of 1000 packages in a day and a fleet size of five eVTOLs. We can see that NSGA2-GG-LS clearly outperforms the three other variants: for each PI, the NSGA2-GG-LS frontier always spans longer, is closer to the origin, and have more solutions. The worst frontier is produced by the basic NSGA2, which confirms the value of having the guided generation of initial solutions and the local search heuristics.

Two additional observations are worth noting. First, between NSGA2-GG and NSGA2-LS, NSGA2-GG finds solutions with greater reduction in community noise impact, whereas NSGA2-LS yields solutions with lower total shipping cost. Second, an NSGA2-LS frontier can have more solutions than an NSGA2-GG frontier. These observations are not surprising and can be explained. For the first observation, recall that when performing guided generation of initial solutions, a range of different weights are assigned to shipping cost and community noise (σ values in Eq. (31)). Thus, the initial solutions contain those that emphasize community noise impact. In contrast, local search heuristics focus more on reducing shipping cost. As a result of this focus, the resulting frontiers can have solutions with low shipping cost. For the second observation, it is related to the fact that guided generation of initial solutions is performed only once at the beginning of NSGA2-GG-LS. As opposed to that, local search is performed at every iteration, which means more exploration of the solution space leading to a greater number of solutions identified on the frontier.

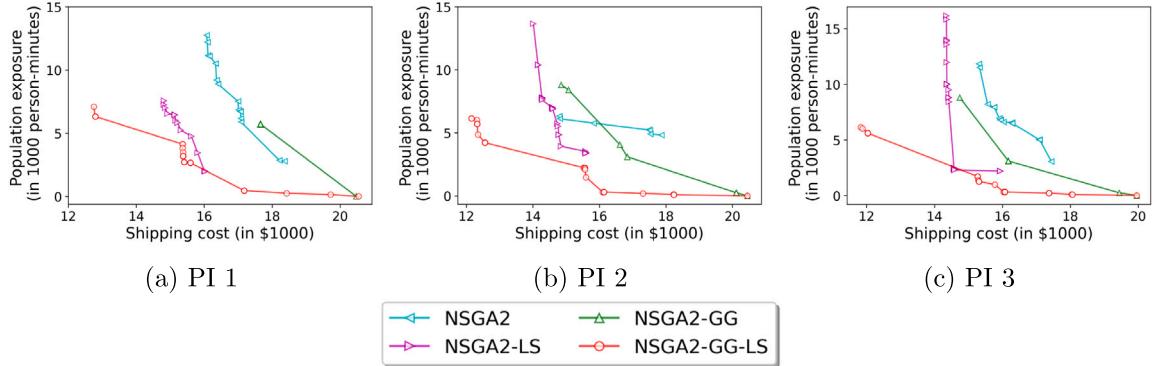


Fig. D.14. Pareto frontiers generated by NSGA2-GG-LS and its variant algorithms.

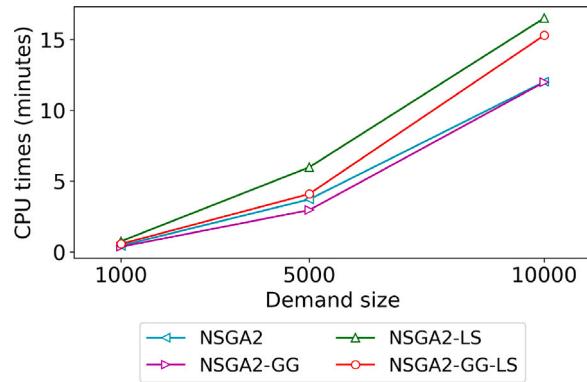


Fig. D.15. Computation time using NSGA2-GG-LS and its variants (each point is an average over five randomly generated PIs).

As NSGA2-GG-LS outperforms NSGA2, NSGA2-GG, and NSGA2-LS in terms of the Pareto frontier quality, we also want to know the difference in computation time. For this, 15 randomly generated PIs of three problem sizes are solved: five PIs with 1000 packages, another five with 5000 packages, and the remaining five with 10,000 packages. For comparison consistency, the same fleet size of 25 eVTOLs is used. For each problem size, the average computation time (CPU time) by the four algorithms is reported in Fig. D.15. We find that the computation time increase is more substantial when incorporating local search than incorporating guided generation of initial solutions, which again can be explained by the fact that the guided generation of initial solutions is performed only once while the local search is performed at every iteration. Another interesting observation is that guided generation of initial solutions can save computation time (by comparing NSGA2-GG with NSGA2, and NSGA2-GG-LS with NSGA2-LS). This is because when applying guided generation, better-quality initial solutions are generated. Consequently, the needed efforts for further solution improvements can be reduced compared to starting with randomly generated initial solutions. For NSGA2-GG-LS, it takes only 14 min for the largest demand size. Considering that our problem is an operation planning problem, this amount of computation time will be acceptable.

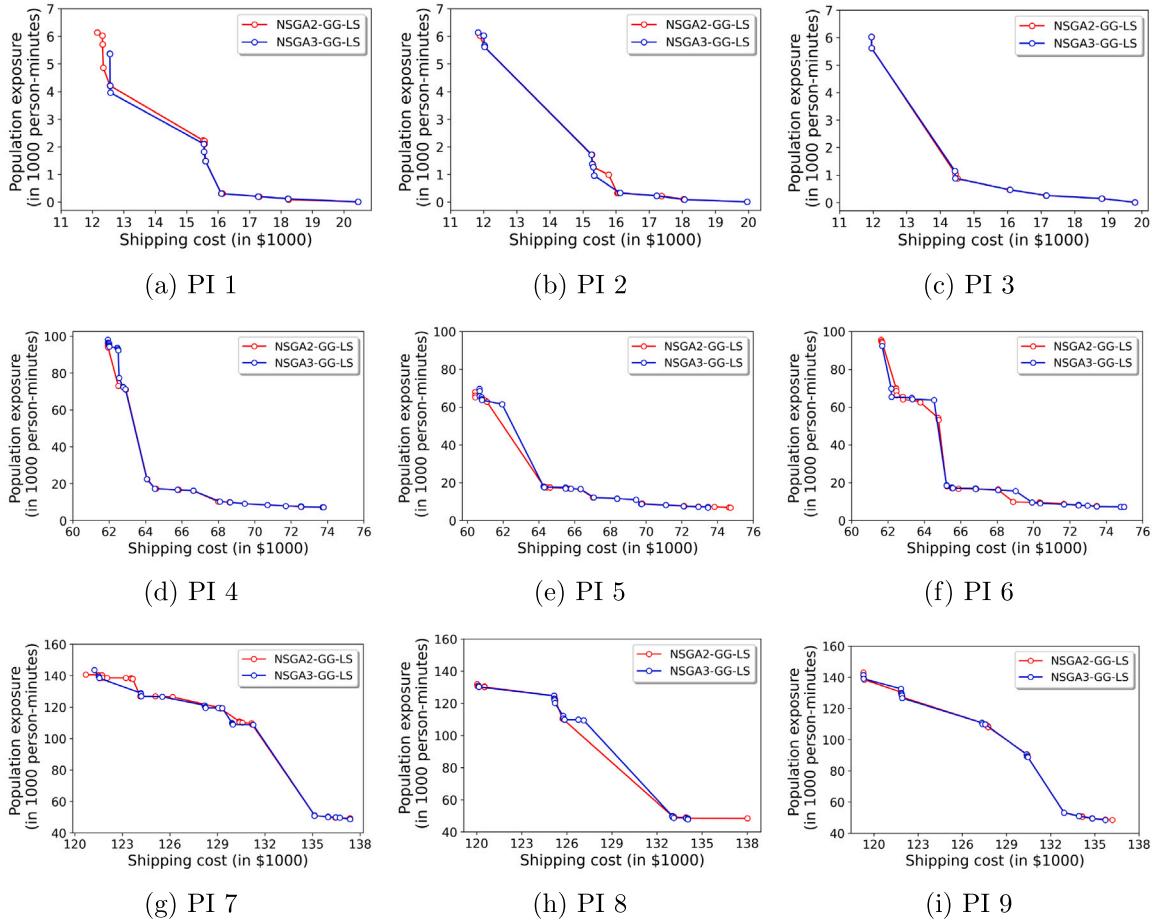


Fig. E.16. Pareto frontiers obtained by NSGA2-GG-LS and NSGA3-GG-LS for the nine PIs in Fig. 8.

Appendix E. Comparison of NSGA2-GG-LS with NSGA3-GG-LS

In this appendix, we compare NSGA2-GG-LS with NSGA3-GG-LS, which differs from NSGA2-GG-LS by replacing NSGA2 by NSGA3. NSGA3 is a recent extension of NSGA2 proposed by Deb and Jain (2013), intended for handling multi-objective optimization problems with more than three objectives. The main distinction of NSGA3 from NSGA2 is the selection mechanism of elite solutions. Instead of computing the crowding distance of a solution as in NSGA2, NSGA3 employs a reference point-based approach to maintain diversity in the solution population. Due to the use of the reference point-based approach, the computation time of NSGA3 is typically greater than NSGA2. On the other hand, for tackling bi-objective optimization, NSGA3 is not necessarily superior to NSGA2, as has been shown in the literature (e.g., Liu et al. (2020), dos Santos et al. (2024)). Our numerical comparison results echo these general descriptions. In Fig. E.16, we present the Pareto frontiers obtained by NSGA3-GG-LS and NSGA2-GG-LS for the nine PIs as in Fig. 8. We observe that the frontiers are very close to each other for each of the nine PIs. This is not surprising, as our problem has only two objectives. Thus, the main strength of NSGA3, which is the ability to efficiently handle multi-objective optimization problems with more than three objectives, is not manifested.

Table E.3 shows that NSGA3-GG-LS in general requires more computation time than NSGA2-GG-LS. The difference in computation time also increases with problem size. For the problem size of 1000 packages and five eVTOLs (PIs 1-3), the average computation time is 27.1 s for NSGA2-GG-LS vs. 28.1 s for NSGA3-GG-LS. For the problem size of 5000 packages and 20 eVTOLs (PIs 4-6), the average computation time is 216.3 s for NSGA2-GG-LS vs. 223.7 s for NSGA3-GG-LS. For the problem size of 10,000 packages and 25 eVTOLs (PIs 7-9), the average computation time is 678.3 s for NSGA2-GG-LS vs. 707.3 s for NSGA3-GG-LS.

Table E.3
Computation time (in seconds) of NSGA2-GG-LS and NSGA3-GG-LS for the nine PIs in Fig. 8.

| | NSGA2-GG-LS | NSGA3-GG-LS |
|-----|-------------|-------------|
| PI1 | 26.4 | 26.7 |
| PI2 | 27.6 | 29.0 |
| PI3 | 27.3 | 28.6 |
| PI4 | 220.1 | 213.9 |
| PI5 | 213.1 | 224.7 |
| PI6 | 215.8 | 232.4 |
| PI7 | 676.0 | 675.4 |
| PI8 | 671.9 | 712.7 |
| PI9 | 687.1 | 733.8 |

Appendix F. Comparison of NSGA2-GG-LS with the ϵ -constrained method and semi-exhaustive enumeration

In this appendix, we compare the Pareto frontiers generated by NSGA2-GG-LS, the ϵ -constrained method, and semi-exhaustive enumeration. The latter two methods are expected to yield reasonably good Pareto frontiers. We solve smaller-size problems given that the needed computation time by two alternative methods quickly increases with the problem size. Specifically, we consider a demand level of 100 packages, an eVTOL carrying capacity of 100 lbs, a fleet size of five eVTOLs, and the top five vertiports with the most noise impact for one eVTOL operation (measured in $\frac{1}{|\mathbb{T}|} \sum_{t=1}^{|\mathbb{T}|} N_k^t$). The eVTOL carrying capacity here is near the lower end of carrying capacities of the existing cargo-carrying eVTOL designs shown in Table 1. The main reason for this is to allow the algorithm to sufficiently explore the solution space before finding a Pareto solution (if the eVTOL carrying capacity is set to 500 lbs, then accommodating 100 packages (each weighs 1–10 lbs) may only need one or two eVTOLs, making the problem very easy to solve). We consider only three time stamps in an operation day. Thus, the time interval between two consecutive time stamps is five hours.

For the ϵ -constrained method, we replace the total community noise impact objective by a constraint that the total community noise impact is less than ϵ . With this replacement, the problem has a single objective of minimizing total shipping cost, and is solved by the branch-and-bound algorithm in CPLEX after linearization. In principle, as the value of ϵ varies at a small pace and covers the possible value range for the total community noise impact, the method is expected to yield a Pareto frontier that is true or close to the true Pareto frontier. A further description of the ϵ -constrained method is provided in Appendix F.1.

For the semi-exhaustive enumeration method, we generate all possible flight schedules (i.e., each flight schedule corresponds to a unique sequencing of 1's and 0's in the solution vector V) while respecting the eVTOL fleet availability constraint. Given a flight schedule, we greedily assign packages to the flights in the same manner as described in Algorithm 1. Once the solutions are generated, non-dominated sorting is performed to construct the Pareto frontier. Note that since the package assignment is performed in a greedy fashion, the generated solutions are not exhaustive. The reason for not exhausting all possible solutions is that the number can be prohibitively large even for a small-size problem. A further description of the semi-exhaustive enumeration method is given in Appendix F.2.

Fig. F.17 shows the Pareto frontiers generated by NSGA2-GG-LS, the ϵ -constrained method, and semi-exhaustive enumeration for three randomly generated PIs. In the figure, we also plot a subset of the enumerated solutions (blue crosses) from semi-exhaustive enumeration. The subset contains all the solutions with shipping cost less than \$3500 and community noise impact less than 8000 person-minutes exposure. We only present this subset as plotting all solutions would make the plots congested and messy. As the Pareto frontiers are generated based on all enumerated solutions, we can see that the non-presented solutions are not on the Pareto frontiers under semi-exhaustive enumeration.

Overall, the Pareto frontier generated by NSGA2-GG-LS is almost as good as the frontier generated by semi-exhaustive enumeration. The NSGA2-GG-LS frontier is also very close to the frontier generated by the ϵ -constrained method. This assures us of using NSGA2-GG-LS to effectively seek the Pareto frontier for the eVTOL-based package delivery problem. It is also worth noting that under semi-exhaustive enumeration, solutions with high shipping cost and low community noise impact are very sparse. This is not surprising, as high shipping cost would likely be associated with a large number of dispatched eVTOL flights, while low community noise impact means a small number of dispatched flights. It is difficult to have a solution that meets both.

To further investigate the computational performance, we randomly generate and solve 15 PIs to compare their computation time (CPU time). The results are reported in Table F.4. The inverted generational distance (IGD) of the frontiers generated by NSGA2-GG-LS and semi-exhaustive enumeration with reference to the frontier generated by the ϵ -constrained method are also reported. IGD is used to measure closeness of a frontier to a reference frontier, as shown in Eq. (F.1) (Coello and Cortés, 2005; Riquelme et al., 2015). In the equation, \mathcal{F} denotes the frontier under measurement. \mathcal{F}^* denotes the reference frontier. s denotes a solution on frontier \mathcal{F} . $\text{dist}(s, \mathcal{F}^*)$ denotes the distance between s and frontier \mathcal{F}^* , calculated as the minimum distance between s and any

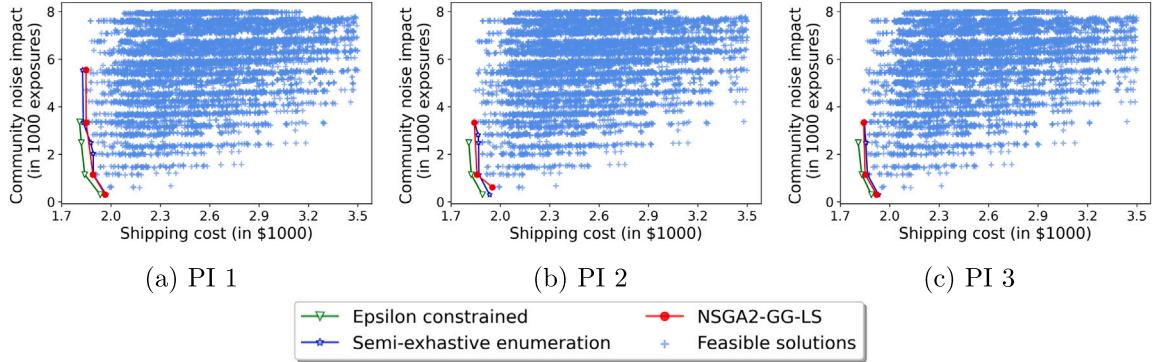


Fig. F.17. Comparison of Pareto frontiers generated by NSGA2-GG-LS, the ϵ -constrained method, and semi-exhaustive enumeration.

Table F.4

Comparison of computational performance of NSGA2-GG-LS, the ϵ -constrained method, and semi-exhaustive enumeration.

| | ϵ -constrained | | Semi-exhaustive | | NSGA2-GG-LS | |
|-------|-------------------------|----------------|-----------------|----------------|-------------|----------------|
| | IGD | CPU time (sec) | IGD | CPU time (sec) | IGD | CPU time (sec) |
| PI 1 | 0 | 2,493.2 | 0.60 | 96.4 | 0.49 | 7.0 |
| PI 2 | 0 | 678.8 | 0.26 | 97.9 | 0.33 | 7.3 |
| PI 3 | 0 | 1,737.5 | 0.37 | 97.7 | 0.43 | 7.3 |
| PI 4 | 0 | 465.8 | 0.54 | 99.2 | 0.51 | 7.0 |
| PI 5 | 0 | 827.7 | 0.47 | 98.3 | 0.49 | 7.1 |
| PI 6 | 0 | 421.0 | 0.54 | 98.7 | 0.55 | 7.1 |
| PI 7 | 0 | 2,968.0 | 0.56 | 99.9 | 0.54 | 7.1 |
| PI 8 | 0 | 435.9 | 0.57 | 98.8 | 0.51 | 7.3 |
| PI 9 | 0 | 964.6 | 0.46 | 96.3 | 0.51 | 7.1 |
| PI 10 | 0 | 1,120.3 | 0.63 | 99.3 | 0.60 | 6.9 |
| PI 11 | 0 | 888.5 | 0.56 | 98.8 | 0.40 | 9.0 |
| PI 12 | 0 | 302.6 | 0.44 | 98.3 | 0.46 | 7.4 |
| PI 13 | 0 | 922.3 | 0.56 | 97.8 | 0.54 | 7.3 |
| PI 14 | 0 | 381.7 | 0.53 | 99.1 | 0.48 | 7.2 |
| PI 15 | 0 | 395.8 | 0.46 | 97.6 | 0.52 | 7.0 |
| Avg. | 0 | 1,000.2 | 0.50 | 98.3 | 0.49 | 7.3 |

point on \mathcal{F}^* . We use the ϵ -constrained method-based frontier as the reference since, as mentioned earlier, the frontier is the true or close to the true Pareto frontier.

$$\text{IGD}(\mathcal{F}, \mathcal{F}^*) = \frac{1}{|\mathcal{F}^*|} \sum_{s \in \mathcal{F}} \text{dist}(s, \mathcal{F}^*) \quad (\text{F.1})$$

Table F.4 shows that NSGA2-GG-LS requires only 6–9 s to solve a PI. In contrast, the needed CPU time by semi-exhaustive enumeration and the ϵ -constrained method are much larger: for semi-exhaustive enumeration, it takes on average 98.3 s to solve a PI; for the ϵ -constrained method, it takes even longer (on average 1000.2 s), along with a significant variation from 302.6 s all the way to nearly an hour. Clearly, NSGA2-GG-LS demonstrates an advantage. The IGD results further show that the frontier generated by NSGA2-GG-LS is as good as using semi-exhaustive enumeration, given that the average IGD values are almost the same. These results provide reassurance that NSGA2-GG-LS can efficiently and effectively produce good-quality frontiers.

F.1. Further description of the ϵ -constrained method

Under the ϵ -constrained method, we remove the community noise impact objective and add the following constraint:

$$\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} y_k^t N_k^t \leq \epsilon \quad (\text{F.2})$$

The model becomes a single-objective nonlinear integer programming model. The nonlinearity arises from $x_{jk}^t z_{1,j}$ and $x_{jk}^t z_{2,j}$ in the objective function (16). They are linearized by introducing auxiliary decision variables $\lambda_{1,jk}^t = x_{jk}^t z_{1,j}$ and $\lambda_{2,jk}^t = x_{jk}^t z_{2,j}$ to the objective function:

$$O'_1 = \min \sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} c_k y_k^t + \sum_{j \in \mathbb{J}} \theta_1 \left(\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} d_{jk} \lambda_{1,jk}^t - \beta d_j^* z_{1,j} \right) + \sum_{j \in \mathbb{J}} \theta_2 \left(\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} t \lambda_{2,jk}^t - t_j^l z_{2,j} \right) \quad (\text{F.3})$$

The following linear constraints need to be added to specify the connection of $\lambda_{1,jk}^t$ with x_{jk}^t and $z_{1,j}$, and of $\lambda_{2,jk}^t$ with x_{jk}^t and $z_{2,j}$:

$$\lambda_{1,jk}^t \leq x_{jk}^t \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.4})$$

$$\lambda_{2,jk}^t \leq x_{jk}^t \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.5})$$

$$\lambda_{1,jk}^t \leq z_{1,j} \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.6})$$

$$\lambda_{2,jk}^t \leq z_{2,j} \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.7})$$

$$\lambda_{1,jk}^t \geq x_{jk}^t + z_{1,j} - 1 \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.8})$$

$$\lambda_{2,jk}^t \geq x_{jk}^t + z_{2,j} - 1 \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.9})$$

$$\lambda_{1,jk}^t = \{0, 1\} \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.10})$$

$$\lambda_{2,jk}^t = \{0, 1\} \quad \forall j \in \mathbb{J}, k \in \mathbb{K}, t \in \mathbb{T} \quad (\text{F.11})$$

Now, the single-objective integer linear programming model has objective (F.3) with constraints (18)–(30), (F.2), and (F.4)–(F.11). The model is solved using by CPLEX multiple times, each time with a different ϵ value. In our numerical experiments, we consider six ϵ values from 1000 to 25,000 in an increment of 4000. Each time after a solution is obtained, we compute the associated community noise impact by $\sum_{t \in \mathbb{T}} \sum_{k \in \mathbb{K}} y_k^t N_k^t$. The Pareto frontier is plotted after all solutions are obtained.

F.2. Further description of the semi-exhaustive enumeration method

To apply semi-exhaustive enumeration, we need to first know how many flight schedules are possible. For the problem size considered in this appendix, we have a fleet of five eVTOLs, five vertiports, and three time stamps. Thus, at each time stamp, a maximum of five flights can be dispatched. In the maximum case, each vertiport will have one eVTOL flight dispatched towards it. Thus in total, a maximum of 15 flights can be dispatched over the three time stamps (given the study area, eVTOL speed, and the time interval between two time stamps of five hours, a dispatched flight can always return to the warehouse and becomes available for the next time stamp). This means that the solution vector V has a dimension of 15. The number of possible flight schedules is $\sum_{|F|=1}^{15} \binom{15}{|F|} = 32,767$. Each flight schedule will yield one solution per the greedy fashion of package assignment.

Once the 32,767 solutions are obtained, the next step is to generate the Pareto frontier. Because directly performing non-dominated sorting of the 32,767 solutions is tedious, a divide-and-conquer strategy is employed. Specifically, we divide the 32,767 solutions into a number of subsets each with a much smaller number of solutions, and apply non-dominated sorting to each subset. Then, the Pareto solutions from each subset are combined to form a new solution set, to which non-dominated sorting is again applied to identify the Pareto frontier, which is the Pareto frontier for the original 32,767 solutions.

To see how the number of solutions by exhaustive enumeration is prohibitively large, let us use a very small example. Consider a problem with only three eVTOLs, three vertiports, two time stamps, and 10 packages. A maximum of $3 \times 2 = 6$ flights can be dispatched. The number of possible flight schedules is $\sum_{|F|=1}^6 \binom{6}{|F|} = 63$. For a given flight schedule, in principle a package can be assigned to any flight in the schedule (for simplicity, we assume that an eVTOL is large enough to carry all packages). Using Stirling numbers of the second kind, which enumerates the number of ways to partition n elements into m non-empty sets, the number of possible solutions given $|J|$ packages and $|F|$ flights can be calculated by:

$$\left\{ \begin{array}{c} |J| \\ |F| \end{array} \right\} = \frac{1}{|F|!} \sum_{i=0}^{|F|} (-1)^i \binom{|F|}{i} (|F|-i)^{|J|} \quad (\text{F.12})$$

Applying Eq. (F.12) to our example and summing over all possible flight schedules, the number of solutions by exhaustive enumeration would be:

$$\sum_{|F|=1}^6 \left(\begin{array}{c} 6 \\ |F| \end{array} \right) \frac{1}{|F|!} \sum_{i=0}^{|F|} (-1)^i \binom{|F|}{i} (|F|-i)^{10} = 983,823 \quad (\text{F.13})$$

which is already very large for such a small problem size.

Appendix G. Nomenclature

Basic notations

| | |
|--------------|--|
| j | Package index |
| \mathbb{J} | Set of packages, $j \in \mathbb{J}$ |
| k | Vertiport index |
| \mathbb{K} | Set of vertiports, $k \in \mathbb{K}$ |
| t | Time stamp index |
| \mathbb{T} | Set of time stamps, $t \in \mathbb{T}$ |
| w_j | Weight of package j |

Notations related to community noise impact quantification

| | |
|----------------|---|
| $L_h(r)$ | Sound level (in dB) on the ground under a noise source (i.e., eVTOL flight) of height h plus a horizontal distance of r from the noise source |
| $I_h(r)$ | Sound intensity (in Watts/m ²) on the ground under a noise source of height h plus a horizontal distance of r from the noise source |
| $P_h(r)$ | Sound pressure (in Pascal) on the ground under a noise source of height h plus a horizontal distance of r from the noise source |
| L_a | Threshold sound level |
| \mathbb{R}_k | Set of rings surrounding vertiport k |
| \mathbb{B}_i | Set of census blocks that have an overlap with ring i |
| ρ_k^t | Parameter representing the fraction of population in the impacted area surrounding vertiport k during the operation of an eVTOL flight that is dispatched at time stamp t |

Parameters in the bi-objective integer programming model

| | |
|------------------------|--|
| Δ | Length of a time interval |
| ϕ_1 | eVTOL turnaround time at a vertiport |
| ϕ_2 | Time for eVTOL check/charging after returning to the warehouse |
| ϕ_3 | Time for package loading at the warehouse for the next flight |
| γ_k | Number of time intervals during which an eVTOL will be unavailable if dispatched to vertiport k |
| α | Minimum remaining time between the departure of a package from the warehouse and its latest delivery time |
| β | Multiplier of the maximum distance between the package destination and the vertiport a package is sent to, with respect to the distance between the package destination and its closest vertiport. |
| t_j^l | Latest time stamp by which a package j needs to leave the warehouse |
| c_k | Cost of an eVTOL flying a round trip between the warehouse and vertiport k , plus the vertiport usage fee |
| θ_1 | penalty parameter for distance violation |
| θ_2 | penalty parameter for time violation |
| $\delta_k^{\tilde{t}}$ | 0–1 indicator to indicate if an eVTOL dispatched towards vertiport k becomes available again \tilde{t} time stamps after its dispatch. |
| T_k | eVTOL one-way travel time between the warehouse and vertiport k |

Notations related to NSGA2-GG-LS

| | |
|-------------------|---|
| s | A solution |
| \mathbb{S}_i | Population of solutions at the beginning of iteration i |
| \mathbb{S}'_i | Population of solutions after NSGA2 operations in iteration i |
| \mathbb{S}''_i | Population of solutions after local search in iteration i |
| \mathbb{S}'''_i | Population of solutions after offspring generation in iteration i |
| G | An integer parameter: $2G$ is the size of \mathbb{S}_0 |
| σ | Weighted parameter when summing the cost- and noise-side attractiveness of vertiport k |
| f | A flight, which is represented by two elements: the vertiport the flight is dispatched towards and at what time stamp: $f = (k_f, t_f)$ |
| F | Set of scheduled eVTOL flights |

References

- Alcock, C., 2022. Amazon invests more in Beta's eVTOL aircraft | FutureFlight. <https://www.futureflight.aero/news-article/2022-06-24/amazon-makes-second-investment-beta-following-alia-evtol-test-flight>. (Accessed 25 December 2022).
- Ale-Ahmad, H., Mahmassani, H.S., 2021. Capacitated location-allocation-routing problem with time windows for on-demand urban air taxi operation. *Transp. Res. Rec.* 2675 (10), 1092–1114.
- Antcliff, K.R., Moore, M.D., Goodrich, K.H., 2016. Silicon valley as an early adopter for on-demand civil vtol operations. In: 16th AIAA Aviation Technology, Integration, and Operations Conference. p. 3466.
- Archer Aviation, 2021. How loud is an eVTOL? <https://archer.com/news/how-loud-is-an-evtol>. (Accessed on 12/08/2022).

- Arntzen, M., Aalmoes, R., Bussink, F., Hoekstra, J., 2015. Noise computation for future urban air traffic systems. In: INTER-NOISE and NOISE-CON Congress and Conference Proceedings, Vol. 250, No. 3. Institute of Noise Control Engineering, pp. 2927–2935.
- Bennaceur, M., Delmas, R., Hamadi, Y., 2022. Passenger-centric Urban Air Mobility: Fairness trade-offs and operational efficiency. *Transp. Res. C* 136, 103519.
- Beyne, E.E., Castro, S.G., 2022. Preliminary performance assessment of a long-range eVTOL aircraft. In: AIAA SCITECH 2022 Forum. p. 1030.
- Bulusu, V., 2019. Urban Air Mobility: Deconstructing the Next Revolution in Urban Transportation-Feasibility, Capacity and Productivity. University of California, Berkeley.
- Bulusu, V., Polishchuk, V., Sedov, L., 2017. Noise estimation for future large-scale small UAS operations. In: INTER-NOISE and NOISE-CON Congress and Conference Proceedings, Vol. 254, No. 2. Institute of Noise Control Engineering, pp. 864–871.
- Casalino, D., van der Velden, W.C., Romani, G., 2019. Community noise of urban air transportation vehicles. In: AIAA Scitech 2019 Forum. p. 1834.
- Choi, J.H., Park, Y., 2022. Exploring economic feasibility for airport shuttle service of urban air mobility (UAM). *Transp. Res. A* 162, 267–281.
- Christian, A.W., Cabell, R., 2017. Initial investigation into the psychoacoustic properties of small unmanned aerial system noise. In: 23rd AIAA/CEAS Aeroacoustics Conference. p. 4051.
- Civil Aviation Authority, 2018. Aircraft Noise and Annoyance: Recent findings. Technical Report CAP 1588, West Sussex, UK.
- Coello, C.A.C., Cortés, N.C., 2005. Solving multiobjective optimization problems using an artificial immune system. *Genet. Program. Evol. Mach.* 6 (2), 163–190.
- Daskilewicz, M., German, B., Warren, M., Garrow, L.A., Bodupalli, S.-S., Douthat, T.H., 2018. Progress in vertiport placement and estimating aircraft range requirements for eVTOL daily commuting. In: 2018 Aviation Technology, Integration, and Operations Conference. p. 2884.
- Deb, K., Jain, H., 2013. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* 18 (4), 577–601.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6 (2), 182–197.
- dos Santos, F., Costa, L.A., Varela, L., 2024. Performance comparison of NSGA-II and NSGA-III on bi-objective job shop scheduling problems. In: International Conference on Optimization, Learning Algorithms and Applications. Springer, pp. 531–543.
- FAA, 2020. Urban Air Mobility (UAM): Concept of Operations v1.0. Technical Report.
- FAA, 2022. Engineering Brief #105: Vertiport Design. Memorandum, <https://www.faa.gov/sites/faa.gov/files/eb-105-veriports.pdf>.
- Fadhl, D.N., 2018. A GIS-Based Analysis for Selecting Ground Infrastructure Locations for Urban Air Mobility (Master's Thesis). Technical University of Munich, [inlangen].
- FedEx, 2022. FedEx plans to test autonomous drone cargo delivery with Elroy Air. <https://newsroom.fedex.com/newsroom/global/elroyair>. (Accessed 25 December 2022).
- Garrett-Glaser, B., 2020. For eVTOL developers, which comes first — the cargo or the passenger? <https://verticalmag.com/features/evtol-developers-which-comes-first-cargo-passenger/>. (Accessed 25 December 2022).
- Garrow, L.A., German, B.J., Leonard, C.E., 2021. Urban Air Mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research. *Transp. Res. C* 132, 103377.
- German, B.J., Daskilewicz, M.J., 2018. Aviation global demand forecast: Model development and ISAAC studies-task 3.7: Concept of operations for ODM VTOL aircraft package delivery.
- German, B.J., Daskilewicz, M.J., Hamilton, T.K., Warren, M.M., 2018. Cargo delivery in by passenger eVTOL aircraft: a case study in the San Francisco bay area. In: 2018 AIAA Aerospace Sciences Meeting. p. 2006.
- Glaab, P., Wieland, F., Santos, M., Sharma, R., Tamburro, R., Lee, P.U., 2019. Simulating fleet noise for notional UAM vehicles and operations in New York. In: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference. DASC, IEEE, pp. 1–10.
- Guisbond, W., 2021. Beta's fully electric Alia aircraft achieves 200+ mile test flight. <https://verticalmag.com/news/beta-alia-aircraft-achieves-200-mile-test-flight/>. (Accessed 22 February 2024).
- Gunady, N.I., Patel, S.R., DeLaurentis, D., 2022. A system-of-systems approach to analyzing future advanced air mobility cargo operations. In: 2022 17th Annual System of Systems Engineering Conference. SOSE, IEEE, pp. 368–373.
- Guo, Y., Hu, M., Zou, B., Hansen, M., Zhang, Y., Xie, H., 2022. Air traffic flow management integrating separation management and ground holding: An efficiency-equity bi-objective perspective. *Transp. Res. B* 155, 394–423.
- Heliport Lighting, 2022. What is a vertiport: Everything you need to know. <https://www.helibrite.com/heliport-lighting/what-is-a-veriport/>. (Accessed 25 December 2022).
- Jahani, H., Chaleshtori, A.E., Khaksar, S.M.S., Aghaie, A., Sheu, J.-B., 2022. COVID-19 vaccine distribution planning using a congested queuing system—A real case from Australia. *Transp. Res. E* 163, 102749.
- Jeong, J., So, M., Hwang, H.-Y., 2021. Selection of vertiports using K-means algorithm and noise analyses for Urban Air Mobility (UAM) in the Seoul Metropolitan Area. *Appl. Sci.* 11 (12), 5729.
- Jia, Z., Lee, S., 2019. Acoustic analysis of a quadrotor eVTOL design via high-fidelity simulations. In: 25th AIAA/CEAS Aeroacoustics Conference. p. 2631.
- Joby Aviation, 2021. Joby aviation analyst day. <https://www.sec.gov/Archives/edgar/data/1819848/000119312521180727/d175333d425.htm>.
- Johnson, W., 1994. Helicopter Theory. Courier Corporation.
- Kafle, N., Zou, B., Lin, J., 2017. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. *Transp. Res. B* 99, 62–82.
- Karimi-Mamaghan, M., Mohammadi, M., Pirayesh, A., Karimi-Mamaghan, A.M., Irani, H., 2020. Hub-and-spoke network design under congestion: A learning based metaheuristic. *Transp. Res. E* 142, 102069.
- Kasliwal, A., Furbush, N.J., Gawron, J.H., McBride, J.R., Wallington, T.J., De Kleine, R.D., Kim, H.C., Keoleian, G.A., 2019. Role of flying cars in sustainable mobility. *Nat. Commun.* 10 (1), 1555.
- Kim, S.H., 2019. Receding horizon scheduling of on-demand urban air mobility with heterogeneous fleet. *IEEE Trans. Aerosp. Electron. Syst.* 56 (4), 2751–2761.
- Klein, V., Steinhardt, C., 2023. Dynamic demand management and online tour planning for same-day delivery. *European J. Oper. Res.* 307 (2), 860–886.
- Kliauskaitė, V., 2021. UPS to buy up to 150 electric cargo aircraft. <https://www.aerotime.aero/articles/27629-UPS-to-buy-electric-aircraft>. (Accessed 25 December 2022).
- Li, S., Egorov, M., Kochenderfer, M.J., 2020. Analysis of fleet management and infrastructure constraints in on-demand Urban Air Mobility operations. In: AIAA Aviation 2020 Forum. p. 2907.
- Licitra, G., 2012. Noise Mapping in the EU: Models and Procedures. CRC Press.
- Lim, E., Hwang, H., 2019. The selection of vertiport location for on-demand mobility and its application to Seoul metro area. *Int. J. Aeronaut. Space Sci.* 20 (1), 260–272.
- Lin, J., Zhou, W., Du, L., 2018. Is on-demand same day package delivery green? *Transp. Res. D* 61, 118–139.
- Liu, D., Huang, Q., Yang, Y., Liu, D., Wei, X., 2020. Bi-objective algorithm based on NSGA-II framework to optimize reservoirs operation. *J. Hydrol.* 585, 124830.
- Liu, M., Lee, C.-Y., Zhang, Z., Chu, C., 2016. Bi-objective optimization for the container terminal integrated planning. *Transp. Res. B* 93, 720–749.
- NASA, 2022. Advanced Air Mobility (AAM) | NASA. <https://www.nasa.gov/aam>. (Accessed 16 December 2022).
- Pipistrel, 2022. Cargo reimagined. <https://www.pipistrel-aircraft.com/air-cargo/>. (Accessed 25 December 2022).
- Polaczyk, N., Trombino, E., Wei, P., Mitici, M., 2019. A review of current technology and research in urban on-demand air mobility applications. In: 8th Biennial Autonomous VTOL Technical Meeting and 6th Annual Electric VTOL Symposium. pp. 333–343.
- Rajendran, S., Srinivas, S., 2020. Air taxi service for urban mobility: A critical review of recent developments, future challenges, and opportunities. *Transp. Res. E* 143, 102090.

- Rajendran, S., Zack, J., 2019. Insights on strategic air taxi network infrastructure locations using an iterative constrained clustering approach. *Transp. Res. E* 128, 470–505.
- Rath, S., Chow, J.Y., 2022. Air taxi skyport location problem with single-allocation choice-constrained elastic demand for airport access. *J. Air Transp. Manag.* 105, 102294.
- Reed, J., 2022. Sabrewing CEO Talks Record-Breaking Cargo Drone Hover Flight. <https://www.aviationtoday.com/2022/10/10/sabrewing-ceo-talks-record-breaking-cargo-drone-hover-flight/>. (Accessed 25 December 2022).
- Rimjha, M., 2022. Urban Air Mobility: Demand Estimation and Feasibility Analysis (Ph.D. thesis). Virginia Tech.
- Rimjha, M., Hotle, S., Trani, A., Hinze, N., 2021a. Commuter demand estimation and feasibility assessment for Urban Air Mobility in Northern California. *Transp. Res. A* 148, 506–524.
- Rimjha, M., Trani, A., Hotle, S., 2021b. Urban Air Mobility: Preliminary noise analysis of commuter operations. In: AIAA Aviation 2021 Forum. p. 3204.
- Riquelme, N., Von Lücken, C., Baran, B., 2015. Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference. CLEI, IEEE, pp. 1–11.
- Rizzi, S.A., Huff, D.L., Boyd, D.D., Bent, P., Henderson, B.S., Pascioni, K.A., Sargent, D.C., Josephson, D.L., Marsan, M., He, H.B., et al., 2020. Urban Air Mobility Noise: Current Practice, Gaps, and Recommendations. Technical Report.
- Sabrewing Aircraft Company, 2021. Air cargo like no other. <https://www.sabrewingaircraft.com/cargo-uav/>. (Accessed 25 December 2022).
- Sabrewing Aircraft Company, 2022. Arabian development & marketing company inks deal with sabrewing aircraft to purchase 53 “Rhaegal-A” VTOL air cargo drones. <https://www.sabrewingaircraft.com/arabian-development-marketing-company-inks-deal-with-sabrewing-aircraft-to-purchase-53-rhaegal-a-vtol-air-cargo-drones/>. (Accessed 25 December 2022).
- Schmähl, M., Rieger, C., Speck, S., Hornung, M., 2022. Semi-empiric noise modeling of a Cargo eVTOL UAV by means of system identification from flight noise measurement data. *CEAS Aeronaut. J.* 13 (1), 85–96.
- Shihab, S.A.M., Wei, P., Ramirez, D.S.J., Mesa-Arango, R., Bloebaum, C., 2019. By schedule or on demand?-a hybrid operation concept for Urban Air Mobility. In: AIAA Aviation 2019 Forum. p. 3522.
- Sripad, S., 2022. Techno-Economics of Battery Electric Mobility & Energy Storage Systems (Ph.D. thesis). Carnegie Mellon University.
- Sripad, S., Viswanathan, V., 2021. The promise of energy-efficient battery-powered urban aircraft. *Proc. Natl. Acad. Sci.* 118 (45), e2111164118.
- Statista, 2020. Quarterly share of e-commerce sales of total U.S. retail sales from 1st Quarter 2010 to 2nd Quarter 2020. <https://www.statista.com/statistics/187439/share-of-e-commerce-sales-in-total-us-retail-sales-in-2010/>. (Accessed 20 August 2023).
- Straubinger, A., Rothfeld, R., Shamiyah, M., Büchter, K.-D., Kaiser, J., Plötzner, K.O., 2020. An overview of current research and developments in Urban Air Mobility – Setting the scene for UAM introduction. *J. Air Transp. Manag.* 87, 101852.
- Tipagornwong, C., Figliozzi, M., 2014. Analysis of competitiveness of freight tricycle delivery services in urban areas. *Transp. Res. Rec.* 2410 (1), 76–84.
- US Census Bureau, 2022. Explore census data. <https://data.census.gov/cedsci/>. (Accessed 04 July 2022).
- US Department of Transportation, 2017. Federal highway administration, national household travel survey. <https://nhts.ornl.gov/>. (Accessed 28 November 2022).
- Vascik, P.D., Hansman, R.J., 2018. Scaling constraints for Urban Air Mobility operations: Air traffic control, ground infrastructure, and noise. In: 2018 Aviation Technology, Integration, and Operations Conference. p. 3849.
- Vertical Flight Society, 2022. ARC Aerosystems Starling Cargo (concept design). <https://evtol.news/samad-starling-cargo>. (Accessed 25 December 2022).
- Willey, L.C., Salmon, J.L., 2021. A method for urban air mobility network design using hub location and subgraph isomorphism. *Transp. Res. C* 125, 102997.
- Yedavalli, P., Mooberry, J., 2019. An assessment of public perception of urban air mobility (UAM). *Airbus UTM Defin. Future Ski.* 2046738072–1580045281.