Optimal Batching and In-Building Delivery Routing with Capacitated Residential Parcel Lockers

Dipayan Banerjee[1] Ignacio Erazo[2]

Abstract. Residential parcel lockers (RPLs), unlike their public counterparts, facilitate secure parcel delivery to occupants of private apartment and condominium buildings in urban areas. In this work, we consider the perspective of a last-mile parcel carrier that has access to an RPL in the lobby of a high-rise residential building. Motivated by growing e-commerce demand, we assume that the number of parcels to be delivered on a particular day exceeds the number of RPL compartments allocated to the carrier; parcels that cannot be delivered to the RPL must be delivered directly to residents' doorsteps on upper floors of the building. Using information about the building's floor layouts and expected elevator waiting times, we seek to optimally group the parcels into batches, determine which parcels to deliver to the RPL, and route the driver through the building to directly deliver the remaining parcels. Under mild assumptions, we first derive a polynomial-time algorithm for determining the optimal sequence of floor visits given a set of parcels designated for direct delivery. We then leverage this algorithm to develop a branch-and-price solution approach capable of solving realistically sized instances of the full problem to optimality. Additionally, we propose and analyze two intuitive heuristics that provide greater transparency and operational simplicity. Our computational experiments suggest that intelligent use of an RPL — whether via exact optimization or heuristics — can entail significant time savings both relative to myopic use of an RPL and relative to the absence of an RPL.

Keywords. last-mile delivery, parcel lockers, column generation, branch-and-price

---

1. Introduction

The duration of a last-mile parcel delivery route can be separated into two main components. The first of these components is the total time spent driving between delivery locations. The second of these components encompasses all activities involved in delivering parcels after parking the vehicle near a delivery location, occasionally referred to as the 'last yard' of the last mile (Song et al. 2019). When delivering a parcel to a suburban or rural single-family dwelling, these delivery activities typically follow a straightforward pattern: the driver retrieves the parcel from the vehicle, walks a very short distance to the building, leaves the parcel by the door, confirms completion of the delivery (e.g., by taking a picture), and returns to the parked vehicle. This allows for simple estimates of per-delivery 'service times' when optimizing the total duration of last-mile delivery routes in suburban and rural areas. For example, Stroh et al. (2022) assume a flat service time of 1.5 minutes per delivery in suburban Atlanta.

[1]Quinlan School of Business, Loyola University Chicago. Equal contribution.
[2]Amazon Fulfillment Technologies & Robotics. Equal contribution.

Historically, optimizing the driving time or distance of delivery routes — via the study of classical traveling salesman problem (TSP) and vehicle routing problem (VRP) variants — has been a primary focus of logistics research. In practice, however, miscellaneous non-driving activities comprise a significant proportion of delivery route durations in urban areas. Data from London and Seattle suggests that this proportion can be as high as 60–80% (Allen et al. 2018, Dalla Chiara et al. 2021). As such, recent approaches to route optimization by Martinez-Sykora et al. (2020), Reed et al. (2022), and others explicitly model the process of a delivery driver walking between their parked vehicle and nearby delivery locations.

Parcel delivery to high-rise residential buildings in densely-populated urban centers can pose additional practical challenges even after the driver arrives at the building on foot. Interior layouts and access policies differ from building to building, potentially causing delays and confusion for drivers. To meet quotas, drivers occasionally opt to leave all of a building's parcels unattended in near the entrance of the building. Perhaps unsurprisingly, this practice raises security concerns due to frequent reports of parcel theft from apartment lobbies (e.g., Broussard 2023, Havens-Bowen 2022, Seyd 2023). In recent years, electronic *residential parcel lockers* (RPLs) have begun to emerge as a convenient alternative to facilitate delivery in urban areas.

Unlike a *public parcel locker*, which is located in a publicly accessible area and can be used to make deliveries to any willing customer, a RPL is dedicated to serving the residents of one particular apartment or condominium building. When a RPL is available for use at a building, a driver places each recipient's parcel into its own RPL compartment, where it is stored securely until it is retrieved with a private access code. This allows drivers to securely deliver parcels to several recipients at the same building in quick succession. Recipients are provided a set amount of time after delivery to retrieve their parcel; afterwards, charges are incurred (Bajaj 2023b) or the parcel is returned (Bajaj 2023a). Firms providing RPL service to residential buildings include Package Concierge and LuxerOne in North America, Parcel Pending in North America and Europe, and Groundfloor in Australia. Amazon also provides RPLs to residential buildings in some countries; this service is distinct from public Amazon pickup points. These Amazon RPLs may or may not be used by non-Amazon carriers at the discretion of building managers (Amazon, Inc. 2024).

Empirical research has demonstrated the utility of RPLs to recipients and delivery firms (Mohri et al. 2024b, Ranjbari et al. 2023a). However, the benefits of RPLs — security for parcel recipients and streamlined operations for delivery drivers — are only realized if a locker compartment is available (i.e., empty) when a driver attempts to deliver a parcel. This is a particular issue when the number of parcels to be delivered is atypically high due to retail events such as Black Friday, Cyber Monday, or Amazon Prime Day.

Additionally, projections suggest that the volume of e-commerce parcel deliveries will continue to increase in the near future (King 2024), potentially rendering the capacity of many previously installed RPLs to be insufficient. When no RPL compartments are available, drivers may place excess parcels near the locker, which again leaves those parcels vulnerable to theft. Another option may be to leave excess parcels in the care of building staff. However, building managers may be unwilling or unable to dedicate space and labor to the time-consuming tasks associated with parcel storage and fulfillment (Kusisto 2015, Pimentel 2022).

A third option that reduces theft risk and is favored by some building managers — including Camden Property Group (Kusisto 2015) and others with whom the authors have personal experience — is to allow building access to delivery drivers to ensure that excess parcels are delivered directly to recipients doorsteps. The drawback of this approach is the need for delivery drivers to spend significant time traveling on foot through the building; this in-building travel time includes the time spent waiting on elevators to arrive, riding the elevator as it moves between floors, and walking through individual floors where recipients' residences are located. This motivates the need for careful optimization of RPL usage: to minimize the time spent delivering parcels, which parcels should be placed in the RPL, and how should drivers be routed within the building to deliver excess parcels? Nevertheless, optimization-based research incorporating RPLs is relatively scarce in contrast to the multitude of studies that optimize aspects of public parcel lockers.

In this work, we take the perspective of a last-mile carrier that sends a driver to deliver a set of small parcels to residents of a high-rise residential building equipped with a RPL. The carrier has knowledge of the number of RPL compartments available to them, which is less than the number of parcels to be delivered. The carrier seeks to minimize the expected length of time that the driver spends delivering the building's packages (i.e., the operation's expected makespan). To this end, we study the linked problems of determining which parcels are placed into the RPL and determining how the driver is routed through the building to deliver the remaining parcels to recipients' doorsteps. Because delivering the full set of parcels may require multiple trips between the vehicle and building, we concurrently determine an optimal separation of the parcels into physical batches. We summarize the main contributions of our work as follows.

1. We formally introduce and model the novel optimization problem of in-building delivery time minimization in the presence of a capacitated RPL.
2. We show that, under realistic assumptions on elevators' behavior and the building's layout, the subproblem of optimally routing the driver within the building to deliver a fixed set of parcels can be

solved in polynomial time.

3. We leverage this result to develop a custom branch-and-price algorithm capable of solving realistically sized instances of the full problem to optimality. We also propose and analyze batching heuristics that, while potentially suboptimal, are intuitive and provide initial columns to the branch-and-price algorithm.

4. We use our solution methods to conduct computational studies in which we demonstrate the value of optimized RPL usage and in-building routing compared to random batch creation and ad hoc routing.

A key finding is that suboptimal use of an RPL can produce worse outcomes than optimal batching and routing without an RPL. We argue, therefore, that realizing the full value of RPL installation requires proactive optimization on the part of parcel carriers. Our contributions are intended to complement recent work on improving the efficiency of parcel delivery operations in dense urban areas through optimized curb space management (e.g., Amaya and Reed 2025) and selection of parking locations (e.g., Reed et al. 2026).

The remainder of the paper proceeds as follows. This section concludes with a literature review. Section 2 details the problem setting and parameters. Section 3 analyzes the subproblem of routing the delivery driver through the building. Sections 4 and Section 5 present exact and heuristic solution methods, respectively, for the full problem. Section 6 conducts computational studies. Section 7 provides concluding remarks and directions for future work. Finally, the appendices contain omitted technical details and data.

1.1 Related Work

We briefly discuss relevant prior research on RPLs and in-building delivery routing. Because column generation and branch-and-price for routing optimization are thoroughly discussed elsewhere, including in Section 4 for the particular context of our problem, we omit these topics from this literature review. We instead refer readers to the recent textbook by Desrosiers et al. (2026) and the references therein.

1.1.1 Residential Parcel Lockers

Three recent papers empirically assess the effects of RPLs via field observations. Mohri et al. (2024b) analyze the benefits of RPLs through a case study of high-rise residential buildings in Melbourne. In buildings without RPLs in their study, delivery drivers enter the building and contact recipients. Upon successful contact, the driver and each recipient meet on the ground floor to exchange parcels. Unsuccessful contact attempts require recipients to pick up parcels at a carrier facility. In buildings with RPLs, all parcels are directly placed in compartments instead. The authors' results suggest that RPLs benefit recipients, carriers, and local governments when compared to the alternative strategy of contacting recipients for handoff.

Ranjbari et al. (2023a) analyze the effects of installing a real-world RPL, accessible by any carrier, in the lobby of a high-rise residential building in downtown Seattle. Before the installation of the RPL, delivery drivers "usually left packages in the building lobby and/or walked throughout the building and went to different floors to make doorstep deliveries." The authors track the total time that delivery drivers spend at the building, but do not analyze the specific movements of drivers within the building. Using a difference-in-differences approach, the authors estimate that the presence of an RPL reduces the average time spent by each delivery driver in the building by over four minutes.

Ranjbari et al. (2023b) again study the operations of a RPL in the lobby of a high-rise residential building in downtown Seattle; it is unclear whether this is the same building analyzed in Ranjbari et al. (2023a). In this setting, delivery drivers leave excess parcels in the lobby. The authors collect and analyze data on RPL usage by carriers and recipients, then leverage the empirical data to perform a simulation analysis to choose the best of six candidate RPL configurations. Of these three papers that study RPLs, Ranjbari et al. (2023b) is the only one that seeks to optimize a decision, albeit from the perspective of building managers and between a very limited set of potential design alternatives.

In contrast, recent years have seen significant research effort devoted to optimizing operational, tactical, and strategic decisions for public parcel lockers in urban areas. Common topics include optimizing delivery routing to public lockers (e.g., Ghaderi et al. 2022) and selecting public locker locations (e.g., Raviv 2023). Despite utilizing the same underlying physical technology, public lockers have two key differences in comparison to the RPLs that motivate our work. First, public lockers are not dedicated to serving any one specific building; rather, they are accessible to any recipient regardless of their home location. Research suggests that a significant percentage of recipients prefer certain public locker locations, such as train stations, farther from their residences (Lyu and Teo 2022). Second, managers of capacitated public lockers often have some mechanism to preemptively control the volume of parcels arriving at the public locker, such as by dynamically rejecting some delivery requests (Sethuraman et al. 2024) or selectively offering locker pickup to certain recipients (Galiullina et al. 2024). We defer to Janinhoff et al. (2024) and Mohri et al. (2024a) for thorough reviews focused specifically on optimization of public parcel lockers and similar non-residential delivery options. Finally, Mohri et al. (2026) study high-level economic outcomes of different cost- and revenue-sharing models for parcel locker management. These include models in which each residential building operates its own RPL and models in which multiple neighboring residential buildings establish a partnership to jointly operate a central semi-public parcel locker.

1.1.2 In-Building Delivery Routing

While the optimization of elevator control algorithms in tall buildings has been heavily researched for decades (Siikonen 2024), in-building routing problems that require elevator travel have seen little attention. We conjecture that this is due to a simple lack of necessity: until the recent increase in residential parcel delivery volume due to the rapid growth of e-commerce, there was little need for a courier to travel through a high-rise residential building to directly deliver parcels to multiple residents' doorsteps.

Of the few papers that explicitly consider deliveries in multi-floor buildings, three study problem settings that are somewhat related to ours, albeit without the presence of an RPL. Gökalp and Uğur (2019) consider an in-building routing problem in which adjacent floors are linked by one or more "connection points" (staircases and elevators). The authors apply a local search metaheuristic to the problem. However, Gökalp and Uğur (2019) make three assumptions that do not reflect the actual high-rise buildings that motivate our work: (i) each floor of the building is a rectangular Euclidean plane, (ii) connection points are only located on the edges of each floor, and (iii) all recipients are visited directly. Paredes-Belmar et al. (2024) study a similar problem for office buildings. A genetic algorithm is used to solve the problem for at most 11 parcels. Like the previous paper, Paredes-Belmar et al. (2024) require all deliveries to be made directly to recipients. Kim et al. (2025a) propose a routing algorithm for robotic in-building delivery of up to seven parcels; their setting has some overlap with the subproblem we study in Section 3.2. However, their application of a shortest path algorithm does not guarantee optimality if the in-building tour involves more than one parcel.

Other papers consider direct deliveries to recipients in high-rise residential buildings, although in settings that are less similar to ours. Lim et al. (2016) solve an urban "three-dimensional" vehicle routing problem (VRP) where every parcel recipient has both a planar coordinate location and a vertical location corresponding to the recipient's floor level. Delivery drivers travel along a road network between buildings, and drivers take elevators to travel vertically within buildings to deliver each parcel directly to its recipient's doorstep. Of note is that every customer is located in a different building, so no routing optimization is required within buildings. The authors model elevator behavior by assuming a fixed vertical travel speed (varying from 2.5 to 5 m/s between instances) and an expected waiting time of three to six minutes (varying between instances) upon requesting an elevator. Chen et al. (2023) study a similar problem, again with only one customer per building, with additional constraints imposed by delivery time windows. Motivated by high-rise residences in Seoul, Park et al. (2016) study a collaborative logistics problem for delivering parcels to recipients' doorsteps. The authors consider up to five deliveries on each in-building route but no RPL. Most recently, Ezaki et al. (2024) model two approaches for making deliveries directly to high-rise residents: to recipients' balconies via drone and to recipients' doorsteps via elevator. Unlike in our problem setting, parcels are assumed to exogenously arrive to the building one at a time.

2. Problem Setting

A last-mile parcel carrier sends a driver to deliver a set of parcels P to residents of a high-rise residential building. The building is equipped with an RPL located in its lobby near the main entrance of the building. As in Raviv (2023) and Sethuraman et al. (2024), the locker has uniform compartments that can each hold one small-sized parcel. On this particular day, the carrier has c available (i.e., empty) RPL compartments to make deliveries. To ensure safety and security, the carrier prefers to deliver as many of the $|P|$ parcels as possible via the RPL. When delivering a subset $S \subseteq P$ of the parcels to the RPL, the driver incurs a setup time (e.g., for walking to/from the RPL and for logging into the electronic RPL system) of α and a per-parcel service time (e.g., for placing the parcel into a compartment and confirming the delivery) of β for a total duration of $\alpha + \beta|S|$. A notification is sent to parcel recipients after the S parcels have been delivered.

In this work, we are interested in the case when there are fewer RPL compartments available than parcels to be delivered (i.e., $c < |P|$). Any parcels that are not placed in the RPL must be delivered directly to recipients' doorsteps on upper floors of the building, necessitating the driver to travel through the interior of the building on foot and via elevator. Upon arrival at a recipient's residence within the building, the driver places the recipient's parcel at the doorstep of the residence and confirms delivery (perhaps by logging the delivery time and taking a picture of the delivered parcel), incurring a per-parcel service time of γ.

An additional practical constraint is that the driver may not be able to carry all of the building's parcels at once. Prior to the driver's departure from the carrier's depot, the parcels P are physically partitioned into separate containers (i.e., *batches*) with handles for ease of movement. Each container has a capacity of q parcels, and the driver can only carry one container at a time. We assume that the parcels P are partitioned into $m = \left\lceil \big(|P| - c\big)/q \right\rceil$ batches designated for doorstep delivery and $m' = \lceil c/q \rceil$ batches designated for RPL delivery. To reduce the likelihood of human error during the delivery process, no batch contains parcels of both types.The driver must return to the parked vehicle after completing all of the deliveries in a batch to return the container and retrieve the next batch. The carrier's problem, then, is to concurrently determine

(i) a separation of P into doorstep-designated batches $D_1, D_2, \ldots, D_m$ and RPL-designated batches $R_1, R_2, \ldots, R_{m'}$, and

(ii) the driver's in-building route for each doorstep-designated batch D_ℓ.

These decisions depend on the details of the driver's in-building travel routes, which in turn depend on the layout of the building and its elevator(s). The carrier's goal is to minimize the total expected time to deliver the building's parcels (i.e., expected makespan). Our choice of a makespan-type objective is driven by two practical considerations: carriers' aversion to paying costly overtime wages to their employees (see e.g., Guo et al. 2025) and the well-documented occupational health risks of long working hours (Wong et al. 2019). We next model the building and its elevators in greater mathematical detail.

2.1 Building Layout

The building in question is a residential high-rise; although the mathematical results presented in this work may be applicable to a variety of buildings, we are specifically motivated by tower- or skyscraper-style buildings with at least twenty floors/stories. The RPL and the delivery vehicle parking area are located on the building's ground floor, which we refer to as the *lobby*. The lobby is mathematically denoted as floor 0 and is followed by floors $\{1, 2, \ldots, n\}$, where larger floor numbers denote floors of higher elevation. The building does not contain a basement accessible by the driver.

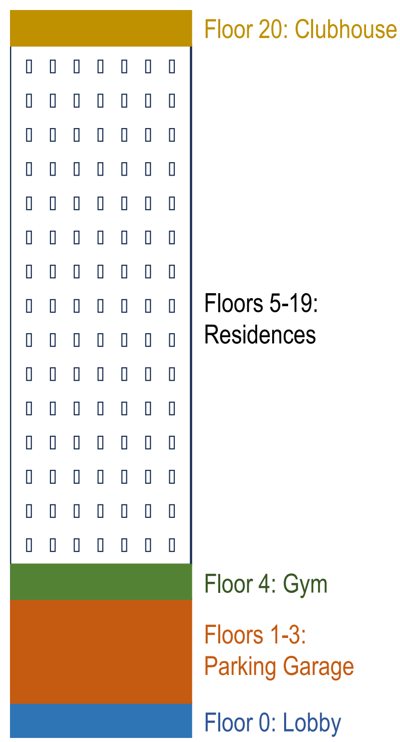

Figure 1: Example residential building, side view

The floors directly following the lobby may be reserved for amenities such as the residents' parking garage, gym, or clubhouse. We also allow for the top floors of the building to be similarly reserved for amenities. We refer to the lobby and amenity floors collectively as *non-residential floors*. The remaining floors contain the building's actual residences and are thus termed *residential floors*. No floors contain both residences and amenities. Figure 1 illustrates a side view of such a building with amenities below and above residential floors.

The floors of the building are all connected by a group of one or more elevators located near the center of each floor. Each residential floor, in particular, contains several residences connected by a hallway. Each hallway is presumed acyclic, facilitating the representation of each residential floor as an undirected tree graph rooted at the elevator group's location. We denote the symmetric walking time between two nodes u and v on the tree graph of floor i as $\tau^i_{u,v}$. Figure 2 illustrates an example of one residential floor's layout (on the left) and its associated graph representation (on the right) from a top-down perspective. The hallway is denoted in dark gray, residential doorsteps (i.e., entryways) are marked in red, and the elevator group is marked with an 'E' on the layout diagram. Note also that a miscellaneous 'service' room, perhaps used for telecommunications equipment or waste, is marked on the layout. The resulting graph representation has a node for each residential doorstep, a node for the elevators, and additional nodes for each location where the hallway branches.

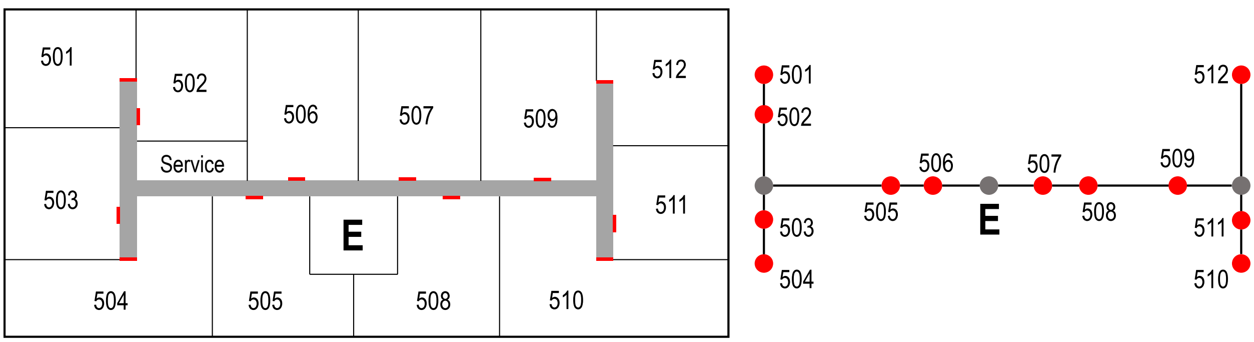

Figure 2: Example floor layout and associated graph representation

2.2 Elevator Behavior

In practice, a significant portion of in-building travel time in high-rise residential buildings is spent (i) waiting on an elevator to arrive once requested and (ii) traveling on an elevator once it arrives. It is therefore helpful to incorporate information about a building's elevators' behavior when estimating in-building travel times and optimizing in-building routing decisions. However, it is unrealistic to leverage detailed knowledge of a residential building's elevator control algorithms for delivery optimization, as "most elevator manufacturers offer their own proprietary algorithms which are mostly treated as trade secrets" (Toll et al. 2020). Instead, to allow for mathematical analysis, we model the driver's elevator travel by defining key parameters and stating (mild) underlying assumptions derived from the authors' real-world experiences.

When the driver seeks to travel to another floor, the driver presses either the 'up' or 'down' call button at the elevator group as necessary. The driver then waits for the arrival of an elevator traveling in the desired direction. In practice, the expected duration of this waiting time is non-negligible, floor-dependent, and direction-dependent. We formally characterize elevator waiting times as follows. The time taken for the elevator to arrive after pressing the 'up' elevator call button, which we refer to as the *upward waiting time*, is a non-negative random variable $W_i^{\uparrow}$ defined by the (cumulative) distribution function $G_i^{\uparrow}(\cdot)$. The time taken for the elevator to arrive after pressing the 'down' elevator call button, which we refer to as the *downward waiting time*, is a non-negative random variable $W_i^{\downarrow}$ defined by the (cumulative) distribution function $G_i^{\downarrow}(\cdot)$. Both of these waiting times are defined to include all miscellaneous activities that occur before the elevator begins to move with the driver onboard, such as the opening of elevator doors, boarding and disembarking of other passengers, and the selection of the destination floor. While $W_i^{\uparrow}$ and $W_i^{\downarrow}$ depend on the trip's origin floor i and desired direction of travel, we emphasize that these values' distributions do not depend on the driver's desired destination floor because the elevator group does not have access to this information until the driver selects a floor once inside an elevator.

Upon boarding the elevator to travel from floor i to floor k, the elevator will travel in the direction selected by the driver at a rate of s seconds per floor until it reaches the destination floor selected by the driver. However, the elevator may stop at intermediate residential floors to drop off or pick up other passengers. For the driver's travel between floors i and k, with $0 \leq i < k-1$, we model intermediate stops as follows. The probability of the elevator stopping at a floor $j \in (i,k)$ when traveling upward from i to k is given by $p_j^{\uparrow}$. Similarly, the probability of the elevator stopping at a residential floor $j \in (i,k)$ when traveling downward from k to i is given by $p_j^{\downarrow}$. Any specific $p_j^{\uparrow}$ or $p_j^{\downarrow}$ value is likely to be small (i.e., much closer to 0 than 1) but not negligible, so there is a significant chance of one or more intermediate stops when the driver travels from the bottom to the top of the building. Whenever the elevator makes an intermediate stop, it stops for an average of ω seconds to allow for boarding and/or disembarkation before continuing its travel in the same direction. Because the elevator is stopped while the driver boards it, we assume that $\mathbb{E}[W_i^{\downarrow}] > \omega$ and $\mathbb{E}[W_i^{\uparrow}] > \omega$ for all i. Combining all of these components, the expected total elapsed time from the moment the driver presses the 'up' elevator call button at floor $i \in \{0, 1, \ldots, n\}$ until the time the driver arrives at floor $k > i$ is modeled as

$$\mathbb{E}[W_i^{\uparrow}] + s(k-i) + \sum_{j=i+1}^{k-1} p_j^{\uparrow}\omega \;=\; \mathbb{E}[W_i^{\uparrow}] + s(k-i) + \omega \sum_{j=i+1}^{k-1} p_j^{\uparrow}\,. \tag{1}$$

Similarly, the expected total elapsed time from the moment the driver presses the 'down' elevator call button at floor k until the time the driver arrives at floor $i < k$ is

$$\mathbb{E}[W_k^{\downarrow}] + s(k-i) + \omega \sum_{j=i+1}^{k-1} p_j^{\downarrow} \, . \tag{2}$$

The first term represents the expected waiting time, the second term represents the elevator's vertical movement, and the third term captures delays due to intermediate stops. Note that there is no possibility of an intermediate stop if floors i and k are adjacent, which would result in the third term being equal to zero.

Finally, in lieu of exact knowledge of a building's elevator control algorithms, we only assume that it is suboptimal for unidirectional elevator travel to 'double back' across floors. Property 1 formalizes this idea.

Property 1. *If the driver starts at floor i and seeks to visit a set of floors J before ending at floor k, where $i < \min\{J\} \leq \max\{J\} < k$, then it is suboptimal to visit the floors J in non-ascending order. Similarly, if the driver starts at floor k and seeks to visit a set of floors J before ending at floor i, where $k > \max\{J\} \geq \min\{J\} > i$, then it is suboptimal to visit the floors J in non-descending order.*

This property is analogous to the non-optimality of self-crossing tours in the metric planar TSP. In practice, this assumption could hypothetically be violated if boarding an elevator traveling in the opposite of the desired direction reduces expected travel and waiting times. However, we presume the elevator's algorithmic control system is sufficiently well-designed to avoid such pathological, non-intuitive behavior.

For clarity in exposition, we henceforth assume that the building is comprised only of residential floors and the lobby. This assumption is without loss of mathematical generality; any residential building whose layout satisfies the structure detailed in Section 2.1 can be mathematically represented as a building with only residential floors and the lobby by appropriately modifying parameters. For example, consider a building for which floors 1–4 are amenity floors. Because the driver will never stop at amenity floors, we can incorporate the travel time and intermediate stopping times associated with these floors when going up or down into $G_0^{\uparrow}(\cdot)$, as the driver's in-building tours are guaranteed to start from the lobby. We can then proceed by assuming that the first residential floor immediately follows the lobby.

3. In-Building Routing Optimization

In this section, we study the subproblem of determining the optimal in-building route, beginning and ending at the lobby, that minimizes the expected time to deliver a given batch of parcels D to recipients' doorsteps. The driver's in-building travel can be decomposed into two independent components that can be optimized separately: routing within each floor (*intra-floor routing*), requiring walking, and routing between floors (*inter-floor routing*), requiring waiting and travel on elevators. The total expected duration of the in-building route is the sum of the inter-floor routing duration, the individual intra-floor routing durations, and the total doorstep delivery service time $\gamma|D|$.

As a notational aside, we use $[i]$ to denote $\{1, 2, \ldots, i\}$ for all positive integers i, and $[0] = \varnothing$. The building's residential floors are therefore $[n]$, and the set of all the building's floors is $[n] \cup \{0\}$. Let $\mathcal{F}(D) = \{F_1, F_2, \ldots, F_k\} \subseteq [n]$ denote the set of residential floors represented in D, with $F_1 < F_2 < \cdots < F_k$. We define $F_0 = 0$ as the lobby for completeness.

3.1 Intra-Floor Routing

Suppose that the driver must deliver one or more parcels to the doorsteps of one or more residences on floor $F_i \in \mathcal{F}(D)$. We seek to minimize the total time that the driver spends walking to deliver parcels on floor F_i. Because the floor is represented as an incomplete graph, and because only a subset of the graph's nodes must be visited, this intra-floor routing problem is an instance of the classical Steiner TSP. Consider the floor depicted in Figure 2 as an example. If the residences on this floor with parcels in D are 501 and 505, the problem is to solve a Steiner TSP on this graph; the nodes required to be visited on the Steiner tour are E, 501, and 505. It is clear that the optimal tour also visits node 506 and the grey node between 502 and 503.

The Steiner TSP on a tree graph is known to be solvable in strongly polynomial time (Erazo and Toriello 2025); trees are naturally submodular graphs (Herer and Penn 1995), and the Steiner TSP on a tree is thus an easily solved submodular minimization problem (see e.g., Schrijver 2000, Iwata et al. 2001). The optimal walking route on each floor is therefore straightforward to compute. We next turn our attention to the inter-floor routing problem, for which an optimal solution method is less readily apparent.

3.2 Inter-Floor Routing

Although the layout of residences on each residential floor can be modeled as an undirected tree graph with edge weights corresponding to walking times, the entire building cannot be modeled as a single, larger undirected tree graph in the same manner. There are two reasons for this, both related to travel on elevators between floors. First, travel times between floors are not additive. For example, travel directly from Floor 10 to Floor 12 incurs an expected duration of $\mathbb{E}[W_{10}^{\uparrow}] + 2s + \omega p_{11}^{\uparrow}$, while travel from Floor 10 to Floor 11 to Floor 12 incurs an expected duration of $\mathbb{E}[W_{10}^{\uparrow}] + \mathbb{E}[W_{11}^{\uparrow}] + 2s$. Second, travel times between floors are not necessarily symmetric: travel from Floor 10 to Floor 11 incurs an expected duration of $\mathbb{E}[W_{10}^{\uparrow}] + s$, while travel from Floor 11 to Floor 10 incurs an expected duration of $\mathbb{E}[W_{11}^{\downarrow}] + s$. Despite the lack of a convenient tree structure, we show in this subsection that inter-floor routing optimization can also be completed in polynomial time.

Formally, beginning and ending at the lobby F_0, we seek a tour between the floors of the building that visits each floor in $\mathcal{F}(D)$ exactly once. The objective is to minimize the total expected time associated with inter-floor travel; this is comprised of waiting, intermediate elevator stops, and elevator movement. We refer to this problem as the *inter-floor TSP* (IFTSP), and we refer to its objective function as the *IFTSP duration*. A solution to the IFTSP (i.e., an inter-floor tour) is denoted by a $(k+2)$-tuple $(F_{x_0}, F_{x_1}, F_{x_2}, \ldots, F_{x_k}, F_{x_0})$, where $F_{x_0} = F_0$ and $(F_{x_1}, F_{x_2}, \ldots, F_{x_k})$ is a permutation of $\mathcal{F}(D) = \{F_1, F_2, \ldots, F_k\}$. The IFTSP can be viewed as the problem of finding which such permutation minimizes the total expected inter-floor travel time. We begin by deriving a fundamental structural property of the IFTSP.

Proposition 2. *In an optimal IFTSP tour, the driver changes direction exactly once after departing the ground floor. Formally, in an optimal IFTSP tour,*

$$F_{x_0} < F_{x_1} < F_{x_2} < \cdots F_{x_{\ell-1}} < F_{x_\ell} > F_{x_{\ell+1}} > \cdots > F_{k-1} > F_{x_k} > F_{x_0}, \tag{3}$$

where $F_{x_\ell} = F_k$ and $F_{x_0} = F_0$.

Proof. Suppose for the purposes of contradiction that, in some optimal IFTSP tour, the driver changes direction more than once after departing the ground floor. Knowing that the driver must visit F_k once on the tour, consider two cases. First, consider the case in which the driver changes direction prior to visiting F_k. Because F_0 is the lowest floor, the driver must travel upward to depart F_0; similarly, because F_k is the topmost floor on the tour, the driver must travel upward to arrive at F_k. Because the driver changes direction prior to visiting F_k, there must be a period of downward travel prior to the eventual upward travel to F_k. This contradicts Property 1.

Second, consider the case in which the driver changes direction after visiting F_k. The driver must travel downward to depart F_k; similarly, the driver must travel downward to arrive at F_0. Because the driver changes direction after visiting F_k, there must be a period of upward travel after the downward travel from F_k. This again contradicts Property 1. □

The implication of Proposition 2 is that the 'backbone' of the optimal tour is an upward journey from F_0

to F_k followed by a downward journey from F_k to F_0. Every other floor in $\mathcal{F}(D) \setminus \{F_k\}$ is visited during either the upward journey or the downward journey. Let $(F_{x_0}, F_{x_1}, \ldots, F_{x_{\ell-1}}, F_{x_\ell}, F_{x_{\ell+1}}, \ldots, F_{k-1}, F_{x_k}, F_{x_0})$ be an optimal IFTSP tour satisfying (3) with $F_{x_0} = F_0$ and $F_{x_\ell} = F_k$. The objective value associated with this tour is then

$$2sF_k \;+\; \sum_{j=0}^{\ell-1} \mathbb{E}\left[W^{\uparrow}_{F_{x_j}}\right] \;+\; \sum_{\substack{j \in [F_k] \setminus \\ \{F_{x_1}, \ldots, F_{x_\ell}\}}} \omega p^{\uparrow}_j \;+\; \sum_{j=\ell}^{k} \mathbb{E}\left[W^{\downarrow}_{F_{x_j}}\right] \;+\; \sum_{\substack{j \in [F_k] \setminus \\ \{F_{x_\ell}, \ldots, F_{x_k}\}}} \omega p^{\downarrow}_j \,. \tag{4}$$

In this expression, $2sF_k$ represents the total time associated with the elevator moving from the lobby to F_k and back. The first summation calculates the total expected waiting time while traveling upward. The second summation calculates the total expected delay due to stopping at intermediate floors during the upward portion of the journey. The third summation calculates the total expected waiting time while traveling downward, beginning at floor $F_k = F_{x_\ell}$. Finally, the fourth summation calculates the total expected delay due to stopping at intermediate floors during the downward portion of the journey.

We can rewrite the expected intermediate stop times to express the objective as

$$\begin{aligned} & 2sF_k \;+\; \sum_{j=0}^{\ell-1} \mathbb{E}\left[W^{\uparrow}_{F_{x_j}}\right] \;+\; \sum_{j=\ell}^{k} \mathbb{E}\left[W^{\downarrow}_{F_{x_j}}\right] \;+\; \sum_{j \in [F_k]} \omega(p^{\uparrow}_j + p^{\downarrow}_j) \;-\; \sum_{j=1}^{\ell} \omega p^{\uparrow}_{F_{x_j}} \;-\; \sum_{j=\ell}^{k} \omega p^{\downarrow}_{F_{x_j}} & (5) \\ &= 2sF_k + \sum_{j=0}^{\ell-1} \mathbb{E}\left[W^{\uparrow}_{F_{x_j}}\right] + \sum_{j=\ell}^{k} \mathbb{E}\left[W^{\downarrow}_{F_{x_j}}\right] + \sum_{j \in [F_k - 1]} \omega(p^{\uparrow}_j + p^{\downarrow}_j) - \sum_{j=1}^{\ell-1} \omega p^{\uparrow}_{F_{x_j}} - \sum_{j=\ell+1}^{k} \omega p^{\downarrow}_{F_{x_j}} \,. & (6) \end{aligned}$$

Rearranging gives

$$\begin{aligned} & 2sF_k \;+\; \mathbb{E}\left[W^{\uparrow}_{F_0}\right] \;+\; \mathbb{E}\left[W^{\downarrow}_{F_k}\right] \;+\; \sum_{j \in [F_k - 1]} \omega(p^{\uparrow}_j + p^{\downarrow}_j) \;+ \\ & \qquad \sum_{j=1}^{\ell-1} \left(\mathbb{E}\left[W^{\uparrow}_{F_{x_j}}\right] - \omega p^{\uparrow}_{F_{x_j}} \right) \;+\; \sum_{j=\ell+1}^{k} \left(\mathbb{E}\left[W^{\downarrow}_{F_{x_j}}\right] - \omega p^{\downarrow}_{F_{x_j}} \right). \end{aligned} \tag{7}$$

The final two summations in this expression clearly illustrate the contribution of each individual floor to the overall objective; these individual contributions depend on whether the floor is visited on the upward or downward part of the tour. The remaining terms represent the fixed contribution to the objective that is determined only by the set of floors to be visited but not by the placement of each floor on the tour.

Given the upward-then-downward tour structure, the IFTSP therefore reduces to the problem of determining, for each $F_i \in \{F_1, F_2, \ldots, F_{k-1}\}$, whether F_i is visited before or after F_k. This problem can be solved efficiently in $\mathcal{O}(k)$ time by performing a simple comparison for each floor, as stated next. The argument, formalized in Appendix A, follows directly from the expression of the objective value in (7).

Proposition 3. *In an optimal solution to the IFTSP, for each floor $F_i \in \{F_1, F_2, \ldots, F_{k-1}\}$, F_i must be visited on the upward journey if $\mathbb{E}[W^{\uparrow}_{F_i}] - \omega p^{\uparrow}_{F_i} < \mathbb{E}[W^{\downarrow}_{F_i}] - \omega p^{\downarrow}_{F_i}$, and F_i must be visited on the downward journey if $\mathbb{E}[W^{\uparrow}_{F_i}] - \omega p^{\uparrow}_{F_i} > \mathbb{E}[W^{\downarrow}_{F_i}] - \omega p^{\downarrow}_{F_i}$.*

3.3 Discussion

A direct consequence of Proposition 3 is that, if elevator behavior is symmetrical in the upward and downward directions, two simple and intuitive IFTSP solutions are optimal. One of these policies begins at the lobby, visits the floors $\mathcal{F}(D)$ in increasing order (i.e., all on the upward journey), then returns to the lobby. The other policy begins at the lobby, travels directly to F_k, then visits the remaining floors in decreasing order (i.e., all on the downward journey), then returns to the lobby. This idea is formalized in Corollary 4.

Corollary 4. *If $\mathbb{E}[W^{\uparrow}_{F_i}] = \mathbb{E}[W^{\downarrow}_{F_i}]$ and $p^{\uparrow}_{F_i} = p^{\downarrow}_{F_i}$ for each $F_i \in \mathcal{F}(D)$, then an IFTSP tour that visits the floors $\mathcal{F}(D)$ in increasing (or decreasing) order is optimal.*

The formal description of elevator behavior in Section 2.2 assumes that the carrier has previously estimated (or obtained estimates for) the various parameters. Prior to the availability of such data, 'increasing order' and 'decreasing order' are reasonable inter-floor routing policies that provide operational simplicity while adhering to the optimal backbone structure of Proposition 2.

As noted earlier, the total expected time to deliver a batch of parcels D is the sum of the IFTSP objective, the individual intra-floor route durations, and the doorstep delivery service time $\gamma|D|$ for all parcels. We henceforth denote this total in-building delivery time as $\mathsf{IBDT}(D)$. The ability to compute $\mathsf{IBDT}(D)$ near-instantaneously by way of (7) facilitates the development of a branch-and-price algorithm and efficient heuristics for the full problem, which we discuss in the following sections.

4. Exact Solution

We formulate the full in-building delivery time minimization problem with RPL availability as follows. Let $\mathcal{B} = \{S \subseteq P : 1 \leq |S| \leq q\}$ be the family of all potential nonempty and feasible doorstep batches. Let x_S be a binary variable that represents the selection of batch S for doorstep delivery with and associated in-building total delivery time of $\mathsf{IBDT}(S)$. Then, the canonical selection/packing master problem can be formulated as the following integer program (IP):

$$\textbf{(MP)} \qquad \min_{x} \quad \alpha m' + \beta c + \sum_{S \in \mathcal{B}} \mathsf{IBDT}(S) x_S \tag{8a}$$

$$\text{s.t.} \quad \sum_{S \in \mathcal{B}: j \in S} x_S \leq 1 \qquad \forall j \in P, \tag{8b}$$

$$\sum_{S \in \mathcal{B}} |S| x_S = |P| - c, \tag{8c}$$

$$x_S \in \{0, 1\} \qquad \forall S \in \mathcal{B}. \tag{8d}$$

The objective function encodes that the full capacity c of the RPL is to be used. Packing constraints (8b) prevent duplicate doorstep delivery service for all parcels. Together with the cardinality constraints (8c), they select exactly the $|P| - c$ distinct parcels designated for doorstep delivery; their complement is therefore exactly the c RPL-designated parcels. Although we can add a constraint of the type $\sum_{S \in \mathcal{C}} x_S = m$ without loss of generality to ensure there are exactly m in-building trips (i.e., the number of doorstep-designated baches), these constraints are redundant due to the definitions of $\mathcal{B}$ and m. We therefore omit such constraints for simplicity of exposition. As an aside, solving solving **(MP)** with $c = 0$ produces a solution that serves the complete building with doorstep delivery, which may serve as a useful point of comparison.

We denote the solution to the full version of this IP — with naïve, complete enumeration of all variables — as `FULL`. However, `FULL` is computationally intractable for many problems of practical size due to exponential growth in the size of $\mathcal{B}$. This motivates the development of a branch-and-price solution approach in which only a subset of columns are generated, which we discuss in detail next.

4.1 Linear Relaxation and LP Duality

We begin by considering the linear relaxation of **(MP)**, with $0 \leq x_S \leq 1$. Define $\mu_j \leq 0$ as the dual variables corresponding to row (8b) in the primal linear program (LP), and define the unrestricted dual variable $\theta \in \mathbb{R}$ associated with row (8c) in the primal LP. As the term $\alpha m' + \beta x$ in **(MP)** is a constant, optimizing its LP relaxation is equivalent to optimizing the following (dual) LP:

$$\max_{\mu, \theta} \quad \sum_{j \in P} \mu_j + (|P| - c)\theta \tag{9a}$$

$$\text{s.t.} \quad \sum_{j \in S} \mu_j + |S|\theta \leq \mathsf{IBDT}(S) \qquad \forall S \in \mathcal{B}, \tag{9b}$$

$$\mu_j \leq 0 \qquad \forall j \in P, \tag{9c}$$

$$\theta \in \mathbb{R}. \tag{9d}$$

4.2 Restricted Master Problem and Pricing

Given $\mathcal{B}$'s exponential size, it is impractical to solve even the LP relaxation of **(MP)** for any large values of $|P| - c$ and q. Instead, we can solve this LP relaxation by leveraging column generation. We start with a subset of doorstep delivery columns $\mathcal{B}' \subset \mathcal{B}$ and solve the LP relaxation of the Restricted Master Problem **(RMP)** (i.e., **(MP)** constrained to columns $\mathcal{B}'$). The **(RMP)** solution is not optimal for the LP unless there are no (optimal) doorstep delivery subsets that we have not yet added to $\mathcal{B}'$; a missing column reflects a violated constraint in the dual, so the pricing problem maximizes

$$\max_{S \in \mathcal{B}} \left\{ \sum_{j \in S} \mu_j + |S|\theta - \mathsf{IBDT}(S) \right\}. \tag{10}$$

A positive optimal solution corresponds to a subset S^* that improves the objective value of the LP relaxation of **(RMP)** and that violates its constraint in the dual (9).

4.3 Pricing Problem as a Compact IP

We next leverage the analysis in Section 3 to formulate the pricing problem as an IP that is compact with respect to the number of parcels to be delivered and the number of 'hallway' edges in the building's floor's graph representations. For each parcel i, let $\ell(i)$ denote the floor at which its recipient is located, and let $r(i) = \ell(i) - 1$. Let E_ℓ denote the set of edges on floor ℓ's graph representation. For each $a \in E_\ell$, let $A_{\ell a} \subseteq P$ denote the parcels whose unique elevator-to-door path uses that edge. We introduce the following binary variables.

- For each $j \in P$, the variable y_j denotes whether parcel j is chosen in the subset.
- For each $\ell \in [n]$, variable f_ℓ denotes whether floor ℓ is visited (i.e., represented among the parcels chosen in the subset). Furthermore, the variable h_ℓ equals 1 if and only if floor ℓ is the highest floor visited, which we refer to as the *peak* of the tour.

- For each $\ell \in [n]$, variables u_ℓ and d_ℓ denote whether floor ℓ is visited on the upward or downward leg, respectively, of the driver's in-building tour corresponding to the chosen subset.

- For each $\ell \in [n]$ and $a \in E_\ell$, the variable $z_{\ell a}$ denotes whether the hallway edge a is used on the driver's in-building tour corresponding to the chosen subset.

The pricing problem's objective is

$$\max_{y,h,f,u,d,z} \sum_{j \in P} \mu_j y_j + \theta \sum_{j \in P} y_j - \widehat{C}, \tag{11}$$

with the subset's in-building doorstep delivery time linearized as

$$\begin{aligned}
\widehat{C} = \gamma \sum_{j \in P} y_j + \mathbb{E}[W_0^{\uparrow}] & \qquad (12a)\\
+ \sum_{\ell=1}^{n} h_\ell \left[2\nu_\ell + \mathbb{E}[W_\ell^{\downarrow}] + \omega \sum_{r=1}^{\ell-1} (p_r^{\uparrow} + p_r^{\downarrow}) \right] & \qquad (12b)\\
+ \sum_{\ell=1}^{n} u_\ell (\mathbb{E}[W_\ell^{\uparrow}] - \omega p_\ell^{\uparrow}) + \sum_{\ell=1}^{n} d_\ell (\mathbb{E}[W_\ell^{\downarrow}] - \omega p_\ell^{\downarrow}) & \qquad (12c)\\
+ \sum_{\ell=1}^{n} \sum_{a \in E_\ell} 2\tau_a^{\ell} z_{\ell a}, & \qquad (12d)
\end{aligned}$$

where $\nu_\ell = \ell s$ denotes the time needed for the elevator to travel between the lobby and floor ℓ assuming no intermediate stops. The first line of the expression corresponds to the total doorstep service time as well as the upward waiting time at the lobby. The second line covers the time spent traveling to and from the peak floor, the downward waiting time at the peak floor, as well as the term following ω that is counter balanced by the terms $\omega p_\ell^{\uparrow}$ and $\omega p_\ell^{\downarrow}$ in the third line to account for the fact that the probability of an intermediate stop at a floor should only considered when it is not being visited directly on the corresponding leg. That is, if we visit floor 4 on the way up, then we should only account for the extra expected time stopping on floor 4 on the way down (i.e., $\omega p_4^{\downarrow}$). The third line also captures the upward and downward waiting times at non-lobby and non-peak floors. The last line covers the intra-floor walking time on the floors that the route visits. Observe the similarities between (12) and the previously derived IFTSP cost (7).

Then, the constraints for the compact pricing IP are

$$1 \leq \sum_{j \in P} y_j \leq q, \tag{13a}$$

$$y_j \leq f_{\ell(j)} \qquad \forall j \in P, \tag{13b}$$

$$f_\ell \leq \sum_{j:\ell(j)=\ell} y_j \qquad \forall \ell \in [n], \tag{13c}$$

$$u_\ell + d_\ell + h_\ell = f_\ell \qquad \forall \ell \in [n], \tag{13d}$$

$$\sum_{\ell=1}^{n} h_\ell = 1, \tag{13e}$$

$$\ell f_\ell \leq \sum_{r=1}^{n} r h_r \qquad \forall \ell \in [n], \tag{13f}$$

$$y_j \leq z_{\ell a} \qquad \forall \ell \in [n], a \in E_\ell,\ j \in A_{\ell a}, \tag{13g}$$

$$z_{\ell a} \leq \sum_{j \in A_{\ell a}} y_j \qquad \forall \ell \in [n], a \in E_\ell. \tag{13h}$$

Constraint (13a) enforces the chosen subset to be non-empty and within the maximum batch capacity q. Constraints (13b) and (13c) link floor visits in both directions; similarly, (13g) and (13h) link graph edges exactly. Constraints (13e) and (13f) define the unique peak of the tour. For a fixed subset of chosen parcels, maximizing (11) subject to these constraints minimizes $\widehat{C}$ and chooses the cheaper direction (up or down) for every non-peak level. We next prove the correctness of this pricing IP formulation.

Proposition 5 (Pricing correctness)**.** *For every $S \in \mathcal{B}$, the minimum of $\widehat{C}$ over the floor/direction/edge variables equals $\mathsf{IBDT}(S)$. Hence* (11) *equals* (10)*, and the IP returns the optimal column for the pricing problem.*

Proof. Fix S. The bidirectional linkage constraints induce the variables $\{f_\ell\}$ to encode the set of floors that S must visit and z to encode the union of the root-to-door paths on each floor. The unique ℓ with $h_\ell = 1$ is a visited floor at least as high as every visited floor, so it is exactly equal to the maximum floor represented in the chosen subset S. Constraints (13d) and (12) then realize the direction at which all the non-peak floors are visited. Finally, $\widehat{C}$ is defined exactly as the total time spent delivering batch S to recipients' doorsteps and is thus equal to $\mathsf{IBDT}(S)$. □

4.4 Branch-and-Price

We solve the original master problem **(MP)** by leveraging branch-and-price (see e.g. Desrosiers et al. 2026). We start by solving the LP relaxation of **(RMP)**, initialized with the columns from the `FIFO` and `FEW` heuristic solutions introduced in Section 5. We then solve the pricing iteratively by adding columns until we obtain the optimal solution for the LP relaxation of **(RMP)**. If this solution is not integer, then we proceed with branching, i.e., creating two children nodes. The procedure terminates when we find an integer solution that we can certify is optimal.

Let $\mathcal{B}'$ be the set of columns for a **(RMP)** node. We define the aggregate delivery-mode value of parcel j to be

$$\hat{y}_j = \sum_{S \in \mathcal{B}' : j \in S} x_S. \tag{14}$$

At integer solutions, $\hat{y}_j = 0$ implies that the parcel's delivery mode is 'RPL' and $\hat{y}_j = 1$ implies that the parcel's delivery mode is 'doorstep', but fractional values do not have a valid interpretation. Branching must therefore settle any such fractional values to create integer solutions. We proceed with branching in two sequential stages:

1. **Mode Branching**. For some $\hat{y}_j \in (0, 1)$, we create a RPL child node with $\hat{y}_j = 0$ and a doorstep child node with $\hat{y}_j = 1$. For the pricing problem, the RPL child node imposes $y_j = 0$. The doorstep child node tightens parcel j's packing constraint (8b) from an inequality to equality, but induces no changes to the pricing structure. Once every $\hat{y}_j$ is integral, the $|P| - c$ parcels with $\hat{y}_j = 1$ form a fixed set of doorstep-designated parcels and their active packing constraints (8b) hold at equality. If the master solution remains fractional, define $\hat{y}_{ab} = \sum_{S \ni a,b} x_S$ for doorstep-designated parcels a, b.

2. **Ryan–Foster Branching** (Ryan and Foster 1981). For a pair of parcels a, b with $0 < \hat{y}_{ab} < 1$, we create two child nodes. One of these nodes is a 'married' child node in which every admissible column contains both or neither parcel, enforced in the pricing problem by the constraint $y_a = y_b$. The other node is a 'separated' child node in which no admissible column contains both parcels, enforced in the pricing problem by $y_a + y_b \leq 1$. These restrictions are structural: the restricted master filters problem incompatible columns and the pricing problem generates only columns that are compatible for the desired child node.

We conclude by formally showing the completeness of this branching approach.

Proposition 6 (Completeness of two-stage branching)**.** *Mode Branching and Ryan–Foster Branching recursively partition all integer solutions of* ***(MP)****. If an LP solution is fractional, either a fractional mode exists or, after all modes are integral, a Ryan–Foster pair with* $0 < \hat{y}_{ab} < 1$ *exists. Hence, an unresolved fractional node always has an eligible branch.*

Proof. For a fractional $\hat{y}_j$, every integer solution has $y_j \in \{0, 1\}$, so the RPL and doorstep children are exhaustive and disjoint. Suppose instead all $\hat{y}$ are integral. The cardinality constraint (8c) fixes exactly $|P| - c$ of them to equal 1, so the residual problem is a set-partitioning problem on that fixed doorstep-designated set of parcels.

It remains to show that a fractional solution has a fractional 'togetherness' value. Suppose, to the contrary, that every $\hat{y}_{ab}$ is either zero or one. Fix a doorstep-designated parcel a. Because $y_a = 1$, the total weight of positive columns containing a is 1. If $\hat{y}_{ab} = 1$, all of that weight also contains b; if $\hat{y}_{ab} = 0$, none of it contains b. Consequently, every positive-weight column containing a has exactly the same membership pattern over the fixed doorstep-designated set. Since **(MP)** has one variable per canonical subset, there is only one such column, and its value must be 1. Repeating this argument for every doorstep-designated parcel shows that all positive variables are one, contradicting fractionality. Thus, some pair must have $0 < \hat{y}_{ab} < 1$.

Every integer partition puts that pair in the same batch or in different batches; the married and separated children encode exactly those exhaustive, disjoint alternatives. Both restrictions are linear in the pricing indicators and preserve the compact pricing problem structure. □

Combining all these components produces a complete branch-and-price solution method that solves **(MP)** to optimality. We henceforth denote this method and its solutions as `BAP`, and we compare its computational performance to `FULL` in Section 6.

5. Heuristics and Random Batching

We next discuss two heuristic approaches to batch creation in Sections 5.1 and 5.2. The motivation for developing these heuristics are twofold. First, from a computational perspective, the heuristic solutions provide warm starts to the naïve IP `FULL` and initial columns to the branch-and-price algorithm `BAP`. Second, the proposed heuristics are transparent and relatively simple to understand; as such, they may be preferred by system managers in practice despite potential suboptimality. Section 5.3 describes benchmark policies in which parcels are randomly separated into batches.

5.1 First-In, First-Out Heuristic

The analysis in Section 3.2 implies that, generally speaking, having the driver visit higher floors entails longer in-building tour durations. Thus, it may be undesirable to have the driver travel to the topmost floors of the building. Based on this insight, we propose the following *first-in, first-out heuristic* (`FIFO`).

Let $e = |P| - c$ denote the number of excess parcels to be delivered to recipients' doorsteps. In `FIFO`, we create batches by grouping parcels in ascending order of their recipients' floors and designating the first (i.e., 'lowest') $|P| - c$ parcels for doorstep delivery. Formally, let $\prec$ denote a strict total order relation on P such that $k \prec \ell$ if the recipient of parcel k lives on a lower floor than the recipient of parcel ℓ. A simple way to create such a total order in practice is to arrange the parcels in ascending order of their recipients' apartment/unit numbers, assuming the common practice of unit numbers beginning with floor numbers (e.g., units 711 and 1234 are located on floors 7 and 12, respectively); parcels with the same recipient can be ordered arbitrarily. Then, with respect to this total order, `FIFO` designates the lowest q parcels for batch D_1, the next q parcels for batch D_2, and so on. After creating the m doorstep-designated batches, the remaining c parcels are similarly partitioned into m' RPL-designated batches. In-building routes for each doorstep-designated batch are computed as in Section 3.2.

5.1.1 Worst-Case Analysis

`FIFO` is intuitive and relatively straightforward to implement. Unfortunately, there exist problem instances for which `FIFO`'s solution performs arbitrarily poorly compared to the optimal solution, underscoring the value of exact optimization. Additionally, `FIFO` can perform arbitrarily poorly even if intra-floor walking time is disregarded. Proposition 7 formalizes these ideas.

Proposition 7. *`FIFO` does not admit a constant-factor approximation guarantee. Furthermore, `FIFO` does not admit a constant-factor approximation guarantee even if $\tau^i_{u,v} = 0$ for all i, u, and v.*

The full proof appears in Appendix A. The underlying intuition is that `FIFO` can perform poorly when the parcels intended for the lowest recipients are spread out across many floors, incurring many occurrences of waiting for the elevator to arrive. It is possible that the `FIFO` solution requires visiting $|P| - c$ distinct residential floors, while the optimal solution visits only one.

5.2 Fewest-Floors Heuristic

Another managerial implication of Section 3.2 is that each additional floor visited by the driver on an in-building tour incurs additional time, as the driver must wait for the elevator after delivering a floor's parcels. Thus, it will often be an inefficient use of the driver's time to visit a floor just to deliver one or two parcels. We next propose the *fewest-floors heuristic* (`FEW`) based on this insight.

The goal of `FEW` is to minimize the total number of times that the driver exits the elevator on a residential floor across all of the driver's in-building tours. For example, if the two in-building tours visit residential floors $\{3, 4, 7\}$ and $\{2, 4, 8, 9\}$, respectively, the total number of times the driver exits the elevator on a residential floor is seven (three on the first tour and four on the second tour). This requires the driver to wait on a residential floor for a total of seven unique periods, even though the total number of unique residential floors visited by the driver is six.

Computing the `FEW` solution requires solving a relatively small IP. With respect to the overall set of parcels P, let the parameter π_i denote the number of parcels intended for recipients on floor i for each $i \in [n]$. For each $i \in [n]$ and $j \in [m]$, we define the integer-valued decision variable $u_{i,j}$ as the number of parcels in batch D_j designated for doorstep delivery to floor i. For each $i \in [n]$ and $j \in [m]$, the corresponding binary variable $v_{i,j}$ tracks whether batch D_j requires visiting floor i. The IP can be formulated as

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} v_{i,j} && && \text{(15a)}\\
\text{s.t.} \quad & u_{i,j} \le \pi_i v_{i,j} && \forall i \in [n], j \in [m], && \text{(15b)}\\
& \sum_{i=1}^{n} u_{i,j} \le q && \forall j \in [m], && \text{(15c)}\\
& \sum_{j=1}^{m} u_{i,j} \le \pi_i && \forall i \in [n], && \text{(15d)}\\
& \sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,j} = \sum_{i=1}^{n} \pi_i - c, && && \text{(15e)}\\
& u_{i,j} \in \mathbb{Z}_{\ge 0},\ v_{i,j} \in \{0,1\} && \forall i \in [n], j \in [m]. && \text{(15f)}
\end{aligned}$$

The objective (15a) minimizes the total number of residential floor visits across all m batches. Constraints (15b) enforce $v_{i,j} = 1$ if $u_{i,j} > 0$. Constraints (15c) enforce the capacity constraints on each batch. Constraints (15d) ensure that at most π_i parcels are delivered to each floor i. Constraint (15e) ensures that the correct number of parcels are designated for doorstep delivery.

There may be several potential optimal solutions to (15) that each visit different floors on their in-building tours. Based on the idea in Section 5.1, we prioritize optimal solutions that visit lower floors by modifying the previous IP. For each $j \in [m]$, let the variable t_j represent the topmost floor visited on batch j's in-building tour. The additional constraints

$$t_j \geq i v_{i,j} \qquad\qquad \forall i \in [n], j \in [m] \tag{16}$$

correctly define these variables. Modifying the objective to

$$\text{lex} \min \quad \sum_{i=1}^{n} \sum_{j=1}^{m} v_{i,j} \;,\; \sum_{j=1}^{m} t_j \tag{17}$$

hierarchically prioritizes the heuristic's primary and secondary goals. Modern commercial solvers, including Gurobi and CPLEX, provide direct functionality for lexicographic optimization without the need for manually weighting objectives.

5.2.1 Worst-Case Analysis

As before, there exist problem instances for which `FEW`'s solution performs arbitrarily poorly compared to the optimal solution, and `FEW` can perform arbitrarily poorly even if intra-floor walking time is disregarded. Proposition 8 formalizes these ideas.

Proposition 8. *`FEW` does not admit a constant-factor approximation guarantee. Furthermore, `FEW` does not admit a constant-factor approximation guarantee even if $\tau^i_{u,v} = 0$ for all i, u, and v.*

The full proof is given in Appendix A. The key idea is that `FEW` can perform poorly when many recipients are clustered together at the top of the building, incurring significant time spent traveling on the elevator and delays from intermediate stops. Taken to the extreme, it is possible that c recipients are distributed between floors 1 and 2, while all remaining recipients are on floor n. In this scenario, `FEW` requires visiting the highest floor n, while the optimal tour only visits floors 1 and 2.

Although Propositions 7 and 8 are seemingly negative results, an important managerial insight is that the two heuristics achieve their respective worst-case performances under different conditions (as detailed in Appendix A). This suggests that the two heuristics might provide a complementary performance; therefore, it may be useful to leverage both and choose the smaller of their solutions' objective values on a case-by-case basis.

5.3 Randomized Batching

We additionally consider randomized batch creation to assess the relative value of our exact and heuristic solution methods. Specifically, we randomly permutate the list of $|P|$ parcels; the first $|P| - c$ parcels in the permutation are designated for doorstep delivery, while the remainder are designated for RPL delivery. The first q parcels in the permutation comprise D_1, the second q parcels comprise D_2, and so on.

For any such random batch assignment, we create three distinct solutions that vary with respect to the driver's in-building travel behavior for a given doorstep delivery batch. The least sophisticated type of solution is `RAND`. in which the driver visits floors in ascending order. Then, at each floor, the driver visits the necessary doorsteps in a random order. The second type of solution, $\texttt{RAND}^{+}$, is similar except that the driver's walking behavior on each floor is optimized rather than random. Finally, $\texttt{RAND}^{++}$ optimizes both elevator travel and walking for each doorstep delivery batch.

As a supplementary benefit of the analysis in Section 3, evaluating the objective value of any of these randomized batching solutions can be done near-instantaneously. We therefore generate one million random solutions of each type for each instance in our computational studies. We compute the mean objective value across these million solutions to serve as an estimator of the average-case performance of each type of solution for a given instance. We also compute the minimum objective value across the million $\texttt{RAND}^{++}$ solutions; this can be viewed, in a sense, as a separate brute-force heuristic in its own right.

6. Computational Studies

In this section, we apply our solution methods to quantify the value of optimizing in-building operations in the presence of a capacitated RPL. Specifically, we seek to quantify the expected time saved by optimally or heuristically batching parcels and optimally routing the delivery driver when compared against randomized batches with less-sophisticated in-building routing. In Section 6.1, we begin by studying medium-sized problem instances with $|P| = 24$ parcels to be delivered in a building with 20 residential floors. We study large-sized problem instances with $P = |72|$ parcels in Section 6.2. Key managerial insights are discussed alongside the results of both settings.

We additionally seek to assess the computational performance of our solution methods, especially the exact branch-and-price approach, as problem instances grow in size. All of the computational experiments in this section are conducted via Python 3.12 on a MacBook Pro with a M3 Max processor and 48 GB of RAM. All linear and integer programs for `FULL` and `BAP` are solved via Gurobi 13.0.

6.1 Medium Instances: 24 parcels

We consider a high-rise residential building with $n = 20$ residential floors above the lobby; the graph representation of each floor has 32 nodes. The carrier seeks to deliver a total of $|P| = 24$ parcels to recipients of the building, of which $c = 8$ parcels will be delivered to the building's RPL. The driver's carrying capacity (i.e., maximum batch size) is set to $q = 8$ parcels. Each batch designated for the RPL incurs a setup time of $\alpha = 140$ seconds, including walking between the lobby and the parked vehicle. Service times for RPL and doorstep delivery are $\beta = \gamma = 20$ seconds per parcel.

We generate 50 random instances under these conditions. For each instance, each parcel's recipient's residential location within the building is selected independently and uniformly at random. Each floor's expected upward and downward waiting times, $\mathbb{E}[W_i^{\downarrow}]$ and $\mathbb{E}[W_i^{\uparrow}]$, are independently chosen uniformly at random between 30 and 60 seconds. Similarly, each floor's probability of and intermediate stop while traveling upward and downward, $p_j^{\uparrow}$ are $p_j^{\downarrow}$, are independently chosen uniformly at random between 1 and 15 percent. The expected duration of any intermediate stop is set to 7 seconds. We also consider a set of 50 identical instances with a larger RPL capacity of $c = 16$ parcels for comparative purposes.

Each instance is first solved via both exact methods, `FULL` and `BAP`, with no limit on the computational runtime. Table 1 reports their average objective values and runtimes across the 50 instances for both $c = 8$ and $c = 16$. Each instance is then solved via both `FIFO` and `FEW`. Table 1 reports the average objective values, optimality gaps (relative to the exact methods), and runtimes of each method separately. Motivated by the discussion in Section 5.2.1, the table also reports results of an additional heuristic that takes the smaller of the `FIFO` and `FEW` objectives on an instance-by-instance basis.

One million random parcel-to-batch assignments are then generated and evaluated with respect to the different in-building travel strategies of `RAND`, `RAND`$^{+}$, and `RAND`$^{++}$. Table 1 reports the mean objective value and optimality gap for each. Note that the runtime reported in each 'Randomized' row is the average time taken (per instance) to evaluate the million random solutions of the corresponding solution type. As discussed previously, the time taken to evaluate any single solution is negligible; e.g., the average time taken per individual `RAND` solution for $q = c = 8$ is approximately 4×10^{-5} seconds. We also report the minimum of each instance's million RAND^{++} objective values, which we interpret as a heuristic that conducts a brute-force search over the solution space. Although we do not discuss it here for brevity, Table B1 in Appendix B contains analogous results for this setting with $q = c = 12$.

Table 1: Summary of computational results, $|P| = 24$ (averages across 50 instances)

		$q = 8,\ c = 8$			$q = 8,\ c = 16$		
		objective (minutes)	% gap	runtime (seconds)	objective (minutes)	% gap	runtime (seconds)
Exact	FULL	27.37	-	28.89	20.86	-	42.01
	BAP	27.37	-	1.11	20.86	-	2.34
Heuristic	FIFO	30.09	9.94	0.00	22.94	9.98	0.00
	FEW	28.19	2.99	0.02	21.36	2.41	0.01
	min(FIFO, FEW)	27.93	2.05	0.02	21.36	2.39	0.01
	RAND^{++}, minimum	28.31	3.46	26.32	20.98	0.60	14.15
Randomized	RAND, mean	42.60	55.64	41.20	29.30	40.46	21.35
	RAND^{+}, mean	34.03	24.33	26.31	25.02	19.94	13.93
	RAND^{++}, mean	33.43	22.14	26.32	24.72	18.50	14.15

The value provided by exact optimization of batching and in-building routing is apparent for both values of c. We can interpret RAND as a proxy for dispatching a driver to the building with arbitrarily constructed batches and no routing guidance. Then, implementing an optimal solution is expected to save the driver an average of over 15 minutes for $c = 8$ and over 8 minutes for $c = 16$. Recall from Section 1 that the non-driving time of urban last-mile delivery routes can be as high as 60–80%. Therefore, the time savings from optimizing these less-studied aspects of parcel delivery are substantial, particularly if a driver is scheduled to visit several such high-rise residential buildings over the course of each working day. Such optimization may become imperative in the near future due to continued growth in parcel delivery volume and an increasing number of high-rise residential buildings being constructed in cities across the US (Parker 2024). Even the use of a straightforward heuristic can produce improved outcomes; FEW performs nearly as well as the exact methods in these instances, and FIFO noticeably outperforms randomized batching without requiring the use of an optimization solver at all.

As as additional point of comparison, we also compute the optimal solution for each instance if all parcels must be delivered to recipients' doorsteps (i.e., $c = 0$). The resulting average objective value is 36.72 minutes. For $c = 8$, this is comparable to the average-case performance of RAND^{++} and significantly better than the average-case performance of both RAND and RAND$^+$. These results illustrate a critical managerial insight: unsystematic use of an RPL can entail worse outcomes when compared to optimized in-building operations without access to an RPL! Therefore, to ensure that all stakeholders realize the benefits of expensive RPL installation and maintenance, it is crucial that carriers apply some degree of systematic planning to their delivery processes in high-rise buildings.

Finally, with respect to computation, FULL requires a reasonable — although non-negligible — runtime to produce its optimal solutions. However, problem instances of this size are near the limit of what FULL can solve before encountering an out-of-memory error. We therefore do not implement FULL in the next subsection's larger instances. On the other hand, our branch-and-price approach is both significantly faster and extremely memory-efficient: an average of only 206 columns are generated by BAP per instance for $c = 8$, and an average of only 318 columns are generated by BAP per instance for $c = 16$.

6.2 Large Instances: 72 parcels

We next consider the same building but with $|P| = 72$ parcels to be delivered, a larger maximum batch size of $q = 18$, and RPL capacities of $c = 18$ and $c = 36$ parcels. The experimental setup is otherwise the same as in Section 6.1, and Table 2 reports the analogous results. Table B1 in Appendix B contains additional results for this setting with $q = c = 12$.

Table 2: Summary of computational results, $|P| = 72$ (averages across 50 instances)

		$q = 18,\ c = 18$			$q = 18,\ c = 36$		
		objective (minutes)	% gap	runtime (seconds)	objective (minutes)	% gap	runtime (seconds)
Exact	BAP	58.97	-	233.91	49.14	-	432.24
Heuristic	FIFO	64.48	9.35	0.00	54.08	10.04	0.00
	FEW	64.00	8.52	0.08	54.01	9.90	0.05
	min(FIFO, FEW)	62.22	5.51	0.08	52.15	6.11	0.05
	RAND^{++}, minimum	73.61	24.82	65.55	58.59	19.22	44.56
Randomized	RAND, mean	114.10	93.49	109.20	87.62	78.31	73.55
	RAND$^+$, mean	85.54	45.06	65.55	68.58	39.56	43.98
	RAND^{++}, mean	83.73	41.99	65.55	67.38	37.12	44.56

Overall, these results continue to demonstrate that optimizing RPL usage and in-building operations provides measurable time savings, albeit with some differences due to the larger scale of these instances.

First and foremost, the value of optimization — whether exact or heuristic — compared to the average-case performance of the randomized heuristics is now significantly greater. Whereas RAND previously performed 40–56% worse on average than the true optimal solution in Table 1, that gap now stretches to 78–93%. Average-case RAND^{++} solutions with optimized in-building travel underperform even the straightforward FIFO batching approach by over 62% in these settings, highlighting the value produced by systematic batch creation. These insights are of practical importance during periods of atypically high parcel volume, such as the e-commerce demand peak between October and December, when last-mile operations are particularly strained; even simple batching heuristics can alleviate pressure on drivers delivering to high-rise residences during such periods.

The relative value of exact optimization over FEW is now greater as well. This suggests that the guiding principle behind this heuristic — reducing the number of unique floor visits — is not alone sufficient to produce near-optimal solutions for larger instances. However, the larger problem scale more clearly illustrates the discussion from Section 5.2.1: FIFO frequently performs well when FEW does not and vice versa, so taking the better of both solutions now provides noticeable improvements over FEW alone.

While the computational runtimes of BAP are no longer negligible, they remain reasonable for a problem that would be solved offline in practice. Beyond the averages in Table 2, we note that BAP terminated in under 15 minutes for all instances with $c = 18$ and in under 24 minutes for all instances with $c = 36$. Memory requirements similarly remain reasonable given the exponential growth in the solution space, with BAP generating an average of approximately 5000 and 9000 columns per instance for $c = 18$ and $c = 36$, respectively. Brute-force search is no longer a viable strategy in a solution space of this size, with even the best of the million RAND^{++} solutions for each instance performing drastically worse on average than even the straightforward FIFO and FEW heuristics.

7. Conclusions and Future Work

This work studied the delivery operations of a last-mile parcel carrier that has access to an RPL in the lobby of a high-rise residential building. Given a restrictive constraint on the number of RPL compartments available, we sought to optimize the linked problems of partitioning a set of parcels into capacitated batches, determining which parcels are placed into the RPL, and determining how a delivery driver should be routed through the building to deliver the remaining parcels to recipients' doorsteps. We leveraged the unique structure induced by the particular nature of three-dimensional in-building travel to develop a custom branch-and-price algorithm and two intuitive heuristics. Our computational experiments demonstrated that detailed modeling and optimization, whether heuristic or exact, of in-building delivery operations can provide significant value compared to ad hoc RPL usage. Because RPLs and in-building delivery optimization have seen relatively little attention in the operations research literature, we believe that there are several opportunities for further refinement, extension, and application of our ideas. We propose three avenues for future academic and practical work related to the contributions of this paper.

First, our problem setting implicitly assumed that the RPL was shared among multiple carriers, and the operator of the RPL (which may be the building's management firm or a third-party RPL management company) exogenously allocated a certain number of RPL compartments to our carrier in advance of the delivery operations modeled in this work. Instead, one may consider the problem of optimally allocating compartments to several carriers (e.g., Amazon, UPS, FedEx, DHL, etc.) in advance of each day's operations from the perspective of the RPL manager. Furthermore, as additional RPL compartments become available during the day due to prior recipients picking up their parcels, should these compartments be made available to carriers on a first-come first-serve basis or via a more sophisticated mechanism? Alternatively, should RPL capacity be sold to carriers for a fee rather than simply being allocated?

Second, this work was specifically motivated by the delivery of small and inexpensive consumer goods. We therefore assumed homogeneity in parcels' sizes and delivery security requirements. That is, any parcel could fit in any RPL compartment, and no parcels specifically required secure RPL delivery due to their high monetary value. One or both of these assumptions can be relaxed if necessary by extending the modeling framework presented in this paper.

Finally, the efficiency of in-building operations in practice relies not only on mathematical optimization but also on drivers' ability to navigate potentially unfamiliar layouts. As an illustrative example in a different in-building context, Kim et al. (2025b) found that Instacart gig workers' order picking improved when provided with a digital tool to support routing inside grocery stores. Therefore, it may be valuable to provide similar navigational aids to guide delivery drivers through optimized in-building routes, particularly in larger buildings or those with unintuitive floor plans. While the development and implementation of such tools would necessarily require information sharing from building management companies, related work on RPL installation and parking space management suggests that cooperation with non-carrier stakeholders can improve the efficiency of urban parcel delivery operations (Amaya and Reed 2025, Mohri et al. 2026).

References

Allen J, Piecyk M, Piotrowska M, McLeod F, Cherrett T, Ghali K, Nguyen T, Bektas T, Bates O, Friday A, Wise S, and Austwick M (2018) Understanding the impact of e-commerce on last-mile light goods vehicle activity in urban areas: the case of London. *Transportation Research Part D*, 61:325–338.

Amaya J and Reed S (2025) Space management policy for urban last-mile parking infrastructure: a demand-oriented approach. *Transportation Research Part E*, 200:104185.

Amazon, Inc (2024) Amazon Apartment Locker. URL https://www.amazon.com/b?ie=UTF8&node=17337376011.

Bajaj S (2023a) Amazon Locker Hubs: how future of apartment living is changing. *Swiftlane*, URL https://swiftlane.com/blog/amazon-lockers/.

Bajaj S (2023b) Parcel Pending lockers: in depth review. *Swiftlane*, URL https://swiftlane.com/blog/parcel-pending-review/.

Broussard L (2023) Man stole more than a dozen packages from apartment building in Mattapan. *NBC Boston*, URL https://www.nbcboston.com/news/local/man-stole-several-packages-from-an-apartment-complex-in-mattapan-police-say/3215527/.

Chen MC, Yerasani S, and Tiwari MK (2023) Solving a 3-dimensional vehicle routing problem with delivery options in city logistics using fast-neighborhood based crowding differential evolution algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 14:10389–10402.

Dalla Chiara G, Krutein KF, Ranjbari A, and Goodchild A (2021) Understanding urban commercial vehicle driver behaviors and decision making. *Transportation Research Record*, 2675(9):608–619.

Desrosiers J, Lübbecke M, Desaulniers G, and Gauthier JB (2026) *Branch-and-Price*. Springer.

Erazo I and Toriello A (2025) Optimizing the trade-off between batching and waiting: subadditive dispatching. *Operations Research*, 73(6):3359–3388.

Ezaki T, Fujitsuka K, Imura N, and Nishinari K (2024) Drone-based vertical delivery system for high-rise buildings: multiple drones vs. a single elevator. *Communications in Transportation Research*, 4:100130.

Galiullina A, Mutlu N, Kinable J, and Van Woensel T (2024) Demand steering in a last-mile delivery problem with home and pickup point delivery options. *Transportation Science*, 58(2):454–473.

Ghaderi H, Zhang L, Tsai PW, and Woo J (2022) Crowdsourced last-mile delivery with parcel lockers. *International Journal of Production Economics*, 251:108549.

Gökalp O and Uğur A (2019) Solving vehicle routing problem for multistorey buildings using iterated local search. *Turkish Journal of Electrical Engineering and Computer Sciences*, 27(5):3516–3531.

Guo H, Qi M, and Qi W (2025) Overworking or outsourcing: integrated prediction-optimization for overtime control with application in last-mile delivery. Preprint, URL https://dx.doi.org/10.2139/ssrn.5054050.

Havens-Bowen A (2022) Thieves keep taking my deliveries from the building lobby. What can I do? *Brick Underground*, URL https://www.brickunderground.com/live/realty-bites-dealing-with-package-delivery-theft-landlords-nyc.

Herer YT and Penn M (1995) Characterizations of naturally submodular graphs: a polynomially solvable class of the TSP. *Proceedings of the American Mathematical Society*, 123:673–679.

Iwata S, Fleischer L, and Fujishige S (2001) A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777.

Janinhoff L, Klein R, Sailer D, and Schoppa JM (2024) Out-of-home delivery in last-mile logistics: a review. *Computers & Operations Research*, 168:106686.

Kim S, Jang H, Ha J, Lee D, Ha Y, and Song YE (2025a) Efficient graph-based multi-story path planning with optimized elevator selection for indoor delivery robots. *Electronics*, 14(5):982.

Kim Y, Knight B, Mitrofanov D, and Xu Y (2025b) Algorithm-enabled decision support and worker learning: a large-scale field experiment. Preprint, URL https://dx.doi.org/10.2139/ssrn.4976809.

King H (2024) Delivery and courier market to hit US $648bn globally by 2030. *Parcel and Postal Technology Interna-*

tional, URL https://www.parcelandpostaltechnologyinternational.com/news/parcels/delivery-and-courier-market-to-hit-us648bn-globally-by-2030.html.

Kusisto L (2015) Web-shopping deluge boxes in landlords: Camden Property Trust takes a tough stand and stops accepting package deliveries. *The Wall Street Journal*, URL https://www.wsj.com/articles/package-gate-big-landlord-blocks-deliveries-after-deluge-of-parcels-1445333401.

Lim SFWT, Zhang AN, Goh M, Ong YS, and Tan PS (2016) Three-dimensional vehicle routing problem for urban last mile logistics: problem formulation and computational analysis. In *2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation*, pp. 252–257.

Lyu G and Teo CP (2022) Last mile innovation: the case of the Locker Alliance Network. *Manufacturing & Service Operations Management*, 24(5):2425–2443.

Martinez-Sykora A, McLeod F, Lamas-Fernandez C, Bektaş T, Cherrett T, and Allen J (2020) Optimised solutions to the last-mile delivery problem in London using a combination of walking and driving. *Annals of Operations Research*, 295(2):645–693.

Mohri SS, Ghaderi H, Farahani RZ, Nassir N, and Thompson RG (2026) Decision models for on-premises parcel lockers service revenue management. *International Journal of Production Economics*, 294:109894.

Mohri SS, Ghaderi H, Van Woensel T, Mohammadi M, Nassir N, and Thompson RG (2024a) Contextualizing alternative delivery points in last mile delivery. *Transportation Research Part E*, 192:103787.

Mohri SS, Nassir N, Thompson RG, and Ghaderi H (2024b) Last-mile logistics with on-premises parcel lockers: who are the real beneficiaries? *Transportation Research Part E*, 183:103458.

Paredes-Belmar G, Latorre-Núñez G, and Bronfman A (2024) Modeling and solving the time-dependent in-building delivery problem in last-mile logistics. *IEEE Access*, 12:11276–11293.

Park H, Park D, and Jeong IJ (2016) An effects analysis of logistics collaboration in last-mile networks for CEP delivery services. *Transport Policy*, 50:115–125.

Parker W (2024) America's apartment buildings are getting bigger and taller. *The Wall Street Journal*, URL https://www.wsj.com/real-estate/americas-apartment-buildings-are-getting-bigger-and-taller-e0d67136.

Pimentel J (2022) As more people shop online, packages overwhelm apartment buildings. *Spectrum News*, URL https://perma.cc/EKS3-J8AF.

Ranjbari A, Diehl C, Dalla Chiara G, and Goodchild A (2023a) Do parcel lockers reduce delivery times? Evidence from the field. *Transportation Research Part E*, 172:103070.

Ranjbari A, Diehl C, Dalla Chiara G, and Goodchild A (2023b) What is the right size for a residential building parcel locker? *Transportation Research Record*, 2677(3):1397–1407.

Raviv T (2023) The service points' location and capacity problem. *Transportation Research Part E*, 172:103070.

Reed S, Campbell AM, and Thomas BW (2022) The value of autonomous vehicles for last-mile deliveries in urban environments. *Management Science*, 68(1):280–299.

Reed S, Campbell AM, and Thomas BW (2026) Should commercial vehicles double park? Optimal parking decisions for last-mile delivery in urban environments. Working paper.

Ryan D and Foster B (1981) An integer programming approach to scheduling. In Wren A, ed., *Computer Scheduling of Public Transport*, pp. 269–280. North-Holland.

Schrijver A (2000) A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355.

Sethuraman S, Bansal A, Mardan S, Resende MGC, and Jacobs TL (2024) Amazon Locker capacity management. *INFORMS Journal on Applied Analytics*, 54(6):455–470.

Seyd J (2023) 'Porch pirate' who scooped Amazon packages from apartments gets 60 days jail. *North Shore News*, URL https://www.nsnews.com/local-news/porch-pirate-who-scooped-amazon-packages-from-apartments-gets-60-days-jail-7849928.

Siikonen ML (2024) Current and future trends in vertical transportation. *European Journal of Operational Research*, 319(2):361–372.

Song J, Wen R, Xu C, and Tay JWE (2019) Service time prediction for last-yard delivery. In *2019 IEEE International Conference on Big Data*, pp. 3933–3938.

Stroh A, Erera A, and Toriello A (2022) Tactical design of same-day delivery systems. *Management Science*, 68(5):3444–3463.

Toll SM, Barbour O, Edwards C, Nichols D, and Day A (2020) Efficient Elevator Algorithm. Technical report, University of Tennessee. URL https://trace.tennessee.edu/handle/20.500.14382/10662.

Wong K, Chan A, and Ngan SC (2019) The effect of long working hours and overtime on occupational health: a meta-analysis of evidence from 1998 to 2018. *International Journal of Environmental Research and Public Health*, 16(12):2102.

Appendix A Omitted Proofs

A.1 Proof of Proposition 3

Proof. Let $(F_{x_0}, F_{x_1}, \ldots, F_{x_{\ell-1}}, F_{x_\ell}, F_{x_{\ell+1}}, \ldots, F_{k-1}, F_{x_k}, F_{x_0})$ be an optimal IFTSP tour satisfying Equation (3) with $F_{x_0} = F_0$ and $F_{x_\ell} = F_k$. For the purposes of contradiction, suppose that a floor $F_{x_i} \in \{F_{x_1}, \ldots, F_{x_{\ell-1}}\}$ on the upward journey has

$$\mathbb{E}[W^{\uparrow}_{F_i}] - \omega p^{\uparrow}_{F_i} > \mathbb{E}[W^{\downarrow}_{F_i}] - \omega p^{\downarrow}_{F_i} \ . \tag{A1}$$

Then, if F_{x_i} were to be visited on the downward journey instead, the objective value would change by

$$\left(\mathbb{E}[W^{\downarrow}_{F_i}] - \omega p^{\downarrow}_{F_i}\right) - \left(\mathbb{E}[W^{\uparrow}_{F_i}] - \omega p^{\uparrow}_{F_i}\right) < 0 \ , \tag{A2}$$

contradicting the optimality of the given tour. An analogous argument for a floor on the downward journey proves the second condition of the Proposition. □

A.2 Proof of Proposition 7

Proof. Let $\bar{W} = \mathbb{E}[W^{\uparrow}_i] = \mathbb{E}[W^{\downarrow}_i]$ and $\bar{p} = p^{\uparrow}_i = p^{\downarrow}_i$ for all floors $i \in [n] \cup \{0\}$. Let $\delta = \max_{i,u,v}\{\tau^i_{u,v}\}$ denote the maximum walking time between any two nodes on the same floor.

Consider an instance with $|P| = 2q$, $c = q$, and $n = q + 1$. Suppose that q of the parcels are intended for a single recipient located on floor $n = q + 1$, while one parcel is intended for a unique recipient on each of the floors $\{1, 2, \ldots, q\}$. The objective value of this solution is no more than

$$\alpha + \beta q + \gamma q + 2s(q + 1) + 2\bar{W} + 2\bar{p}\omega q + 2\delta \tag{A3a}$$

$$= \alpha + 2\bar{W} + 2\delta + 2s + q(\beta + \gamma + 2s + 2\bar{p}\omega), \tag{A3b}$$

corresponding to the strategy of designating the parcels on floor $q + 1$ for doorstep delivery; this serves as an upper bound on the optimal objective value. On the other hand, the `FIFO` solution designates the parcels on floor $q + 1$ for RPL delivery, requiring the driver to visit all of the floors $\{1, 2, \ldots, q\}$. The associated objective value is at least

$$\alpha + \beta q + \gamma q + 2s(q + 1) + \bar{W}(q + 1) + \bar{p}\omega(q - 1) \tag{A4a}$$

$$= \alpha + \bar{W} - \bar{p}\omega + 2s + q(\beta + \gamma + 2s + \bar{W} + \bar{p}\omega). \tag{A4b}$$

Taking limits gives a lower bound on the approximation ratio of

$$\lim_{q\to\infty}\left(\lim_{\bar{W}\to\infty}\frac{\alpha+\bar{W}-\bar{p}\omega+2s+q(\beta+\gamma+2s+\bar{W}+\bar{p}\omega)}{\alpha+2\bar{W}+2\delta+2s+q(\beta+\gamma+2s+2\bar{p}\omega)}\right)=\lim_{q\to\infty}\frac{q+1}{2}=\infty, \tag{A5}$$

as desired. Assuming $\delta=0$ proves the second part of the result. □

A.3 Proof of Proposition 8

Proof. Let $\bar{W}=\mathbb{E}[W_i^{\uparrow}]=\mathbb{E}[W_i^{\downarrow}]$ and $\bar{p}=p_i^{\uparrow}=p_i^{\downarrow}$ for all floors $i\in[n]\cup\{0\}$. Let $\delta=\max_{i,u,v}\{\tau_{u,v}^i\}$ denote the maximum walking time between any two nodes on the same floor.

Consider an instance with $|P|=4$, $c=2$, and $q=2$ with $n\geq 3$. Suppose that two of the parcels are intended for a particular recipient on floor n, one parcel is intended for a recipient on floor 1, and the final parcel is intended for a recipient on floor 2. The objective value of this solution is no more than

$$\alpha+2\beta+2\gamma+4s+3\bar{W}+\bar{p}\omega+2\delta, \tag{A6}$$

corresponding to the strategy of designating the parcels on floors 1 and 2 for doorstep delivery; this serves as an upper bound on the optimal objective value. On the other hand, the `FEW` solution designates the parcels on floor n for doorstep delivery, producing an objective value of at least

$$\alpha+2\beta+2\gamma+2ns+2\bar{W}+2(n-1)\bar{p}\omega. \tag{A7}$$

Taking a limit gives a lower bound on the approximation ratio of

$$\lim_{n\to\infty}\frac{\alpha+2\beta+2\gamma+2ns+2\bar{W}+2(n-1)\bar{p}\omega}{\alpha+2\beta+2\gamma+4s+3\bar{W}+\bar{p}\omega+2\delta}=\infty, \tag{A8}$$

as desired. Assuming $\delta=0$ proves the second part of the result. □

Appendix B Additional Computational Data

Table B1: Summary of computational results, $q = c = 12$ (averages across 50 instances)

		$\lvert P\rvert = 24$			$\lvert P\rvert = 72$		
		objective (minutes)	% gap	runtime (seconds)	objective (minutes)	% gap	runtime (seconds)
Exact	FULL	21.04	-	130.11	-	-	-
	BAP	21.04	-	0.73	69.00	-	154.50
Heuristic	FIFO	23.35	10.99	0.00	74.91	8.55	0.00
	FEW	21.58	2.58	0.01	76.16	10.37	0.29
	min(FIFO, FEW)	21.58	2.56	0.01	73.26	6.17	0.29
	RAND^{++}, minimum	21.11	0.32	17.26	89.59	29.83	83.41
Randomized	RAND, average	32.12	52.66	27.74	134.66	95.16	136.67
	RAND^{+}, average	25.77	22.48	17.21	102.59	48.68	83.54
	RAND^{++}, average	25.34	20.44	17.26	100.37	45.46	83.41