

Robust Planning of Sorting Operations in Express Delivery Systems

Reem Khir*, Alan Erera, Alejandro Toriello

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology, Atlanta, GA, 30332, USA

reem.khir@gatech.edu, {aerera,atorielle}@isye.gatech.edu

July 15, 2022

Abstract

Parcel logistics services play a vital and growing role in economies worldwide, with customers demanding faster delivery of nearly everything to their homes. To move larger volumes more cost effectively, express carriers use sort technologies to consolidate parcels that share similar geographic and service characteristics for reduced per-unit handling and transportation costs. This paper focuses on an operational planning problem that arises in two-stage sort systems operating within parcel transportation networks. In this context, primary sorters perform an initial grouping of parcels into “piles” that are subsequently dispatched when necessary to secondary sorters; there, each pile’s parcels are fine-sorted based on their final destinations and service class for packing into outbound transportation vehicles. Such systems must be designed to handle a high degree of uncertainty in the quantity and timing of arriving parcels, yet must also group and sort the parcels to meet tight departure deadlines. Thus motivated, this paper presents robust planning models that assign parcels to sort equipment while protecting against different sources of demand uncertainty commonly faced by parcel carriers. We demonstrate the computational viability of the proposed models using realistic-sized instances based on industry data, and show their value in providing sort plan alternatives that trade off operational costs and levels of robustness.

Keywords: Logistics, sorting operations, express parcel delivery, robust optimization

1 Introduction

Fast and reliable delivery is an essential element in the success of modern e-commerce, which is growing rapidly and driving tremendous growth in demand for parcel logistics services. The global

*Corresponding author.

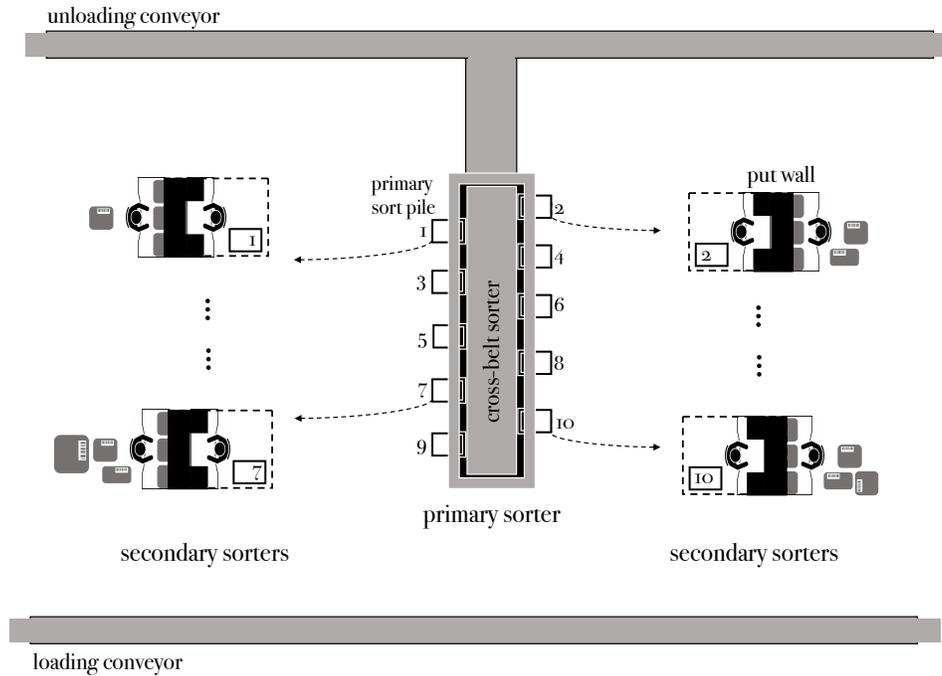
express parcel delivery market was valued at around \$430 billion in 2019, a 13% increase from 2018 (Research & Markets, 2020). This growth trend is expected to continue, with more customers requiring faster delivery of nearly everything to their homes. To move larger volumes more cost effectively, parcel carriers typically operate hub-and-spoke transportation networks that enable parcel flows to be consolidated. Parcel consolidation takes place in sort hubs within these networks, with the goal of grouping parcels that share similar geographic and service characteristics into larger containers to gain economies of scale in their handling and transportation operations.

A key component in parcel sort hub operations is the technology used to carry out the sorting process. There is a wide variety of sort technologies, ranging from fully automated systems, such as cross-belt sorters, tilt-tray sorters and sorting robotics, to semi-automated and manual systems, such as put walls or cabinet-style installations that may require manual handling to complete the sort process. These technologies are typically used within either a *single-stage* or a *multi-stage* sort process, most commonly a two-stage process. A single-stage sort process uses a fully automated sort system designed to have enough sort positions (destinations) and capacity to complete parcel sorting in one step. Well-designed single-stage sort processes are efficient to operate; however, they require large investments in space and equipment and are often not flexible to reconfigure once installed. Two-stage sort processes, on the other hand, are designed to allow for a more flexible setup that is often easier to reconfigure or expand once installed. They can also help reduce parcel travel time in large hubs that operate a complex network of conveyors by eliminating the need for a parcel to travel throughout the complete conveyor system (Boysen, Briskorn, et al., 2019). Contrary to a single-stage sort process, a parcel in a two-stage sort system may go through more than one sort stage before it is considered sorted for its final destination. Two-stage sort systems require careful planning and coordination between the stages to achieve operational efficiencies and meet parcel sort deadlines.

This paper focuses on an operational planning problem that arises in two-stage parcel sort hubs with primary and secondary sorters. Primary sorters perform an initial grouping of parcels into piles that are dispatched as necessary to secondary sorters; there, each pile's parcels are segregated based on their final destinations and service classes for sort and packing before they are sent to the loading area for dispatch. An example of a hub that operates a two-stage sort process is portrayed in Figure 1.

Planning sort operations in this setting includes determining parcel assignment to primary

Figure 1: An example of a two-stage sort process where a cross-belt sorter is used for primary sorting and put walls are used for secondary sorting.



sort pile positions. In some cases, it also involves determining a dispatch schedule of parcels from primary to secondary sort areas to minimize transfer costs and/or better manage floor space. An example setup where the latter decision is applicable includes two-stage sort systems that involve manual transfer operations between the primary and secondary sort areas; see e.g. (Khir et al., 2021). Usually, a relatively small number of sort pile positions are available at the primary stage, while the sortation process must accommodate a large number of parcels with different destinations and service requirements; thus, parcels with multiple destinations and/or deadlines must often be assigned to one primary sort pile position. Parcels assigned to a pile with a single geographic destination and dispatch deadline complete their sort process at the primary sort area, while parcels assigned to piles with different geographic destinations and/or dispatch deadlines are transferred to secondary sorters for detailed final sorting. The efficiency of such a system is largely dependent on how well parcels are assigned to the available primary sort positions, which is the focus of this work. The main challenge is to find an efficient assignment while taking into account (1) the limited number of primary sort positions, (2) the limited downstream capacity for

secondary sorting, and (3) parcel sort time deadlines.

Sort plans are traditionally generated in advance of operations using *expected* demand values, and they are operated and modified every few weeks using updated demand forecasts and consolidation plans (Jarrah et al., 2016; Khir et al., 2021; Novoa et al., 2018). During an actual operational day, however, parcel arrivals can vary significantly from expectations in terms of both volume and timing; thus, operating plans that are based on expected flows may be overly costly in practice or may not meet desired levels of service. To illustrate, consider for example a case where a group of parcels shows up an hour later than their expected arrival time, or a case where more parcels arrive than expected for a particular destination. These variations may cause imbalanced operations across the sort hub; importantly, sorters may not have enough capacity to process all arrivals by their sort deadlines, delaying their ultimate delivery. The frequent changing of sort plans to address fluctuation is not practical, so it is important to generate plans that take into account demand changes while remaining cost-effective and feasible for longer periods of time.

In this paper, we make the following main contributions:

1. Introduce robust optimization models to configure primary sort equipment operating within express parcel delivery sort hubs by explicitly incorporating various sources of demand uncertainty commonly faced by parcel carriers. Using practitioner insights and industry data, we propose robust models that hedge against uncertainty in arrival quantities and/or arrival times that are critical for meeting sort time deadlines.
2. Demonstrate the computational viability of the proposed models using realistic-sized instances based on industry data and show that high-quality solutions can be obtained in relatively short computation times.
3. Show the value of the proposed robust models in providing hub managers with sort plan alternatives that quantify trade-offs between operational costs and different levels of robustness.
4. Extend the model to explore the value of introducing more flexibility to sort operations by allowing pile assignments to change during a sort shift. We show that a single reconfiguration can help reduce the price of robustness, especially during shifts that process small parcel volumes heading to a wide range of destinations.

The remainder of this paper is organized as follows: Section 2 discusses related literature. Section 3 describes the problem and presents the corresponding nominal formulation. Section 4 presents robust formulations that incorporate uncertainty in arrival quantities and/or arrival times. Section 5 illustrates the computational and operational performance of the proposed models using real-sized instances. Finally, Section 6 presents our conclusions.

2 Related Literature

A well-studied stream of research in the context of parcel sort hubs focuses on decisions related to facility location (see Alumur and Kara, 2008 and Farahani et al., 2013 for comprehensive reviews), facility layout (see *e.g.*, Bartholdi and Gue, 2004; Fedtke and Boysen, 2017; Zi and Gao, 2020), and other tactical network design related problems (see *e.g.*, Crainic, 2000 and Baldi et al., 2019). This literature survey focuses on the operational planning of sort systems once design decisions related to hub location, technology selection, and equipment layout and sizing have been fixed. Much of the research on this topic examines inbound and outbound operations that interface with single-stage sort systems. It addresses two primary decision problems, determining the processing schedules of inbound/outbound vehicles (see *e.g.*, Boysen et al., 2017; Chen et al., 2019; McWilliams, 2009; McWilliams and McBride, 2012; McWilliams et al., 2005; Zhou et al., 2021), and the assignment of destination to unloading/loading doors (see *e.g.*, Haneyah et al., 2014; Jarrah et al., 2016; Novoa et al., 2018). Several studies investigate operational policies to improve the throughput and service performance of automated sorters in order fulfillment centers and warehouses (Çeven & Gue, 2017; Gallien & Weber, 2010; Yu et al., 2019). We refer the reader to Boysen, Briskorn, et al., 2019 and Haneyah et al., 2013 for in-depth reviews of related work with a focus on hubs operating fully automated sortation conveyors, a setting that is studied widely in the literature. Only a few studies consider semi-automated or manual sort equipment, such as put walls (Boysen, Stephan, et al., 2019; Khir et al., 2021). The modeling presented in this paper is flexible and can be used for sort hubs operating high- and/or low- tech sort solutions.

Relatively little work has been done on the detailed planning and control of sort operations, especially in the context of multi-stage sort systems. The authors in Briskorn et al., 2017 propose exact dynamic programming approaches to determine the scheduling and routing of shipments in closed-loop tilt tray conveyors such that the total makespan is minimized. Their general

setting considers a one-stage sort system, and therefore, a single destination is to be assigned to one and only one loading station, a key difference from the destination-to-pile assignment problem that arises in two-stage sort systems, which requires many-to-one assignments. The authors in Werners and Wülfing, 2010 focus on optimizing transport activities within parcel sort hubs operated by Deutsche World Post Net. A chance-constrained approach is used to address demand stochasticity and is shown to generate optimal solutions in around 24 hours for real-life instances. Closely related to our research, Novoa et al., 2018 examines the problem of assigning parcel destinations to secondary sort workstations with the objective of improving workload balance in an automated parcel sort hub while explicitly incorporating stochasticity in parcel flows. Parcels are expected to be continuously processed and, therefore, service or time requirements are not explicitly considered in the paper, a key difference between this work and ours. A chance-constrained approach is used to model demand uncertainty in balancing constraints while a robust optimization approach is used to model demand uncertainty in capacity constraints; we use a robust optimization approach similar to the latter in our work. The computational experiments show that high quality solutions can be achieved within one-hour time limit mainly when the robustness level for capacity constraints is less restrictive, *i.e.*, when the uncertainty budget is relatively large, and when the variability between package flows is high. In instances where the robustness level is more restrictive, *i.e.*, when the uncertainty budget is not large, and when there is low variability between parcel flows, the proposed models become less tractable and result in solutions with quite large relative optimality gaps. In our work, we use comparable instances in composition and size and show that the proposed robust models can generate high-quality solutions for various robustness levels in moderate CPU time.

More closely related to our research, the authors in Khir et al., 2021 introduce a sort plan design problem in the context of two-stage sort systems. An optimization-based approach is proposed that decomposes the problem into two subproblems: (1) a primary sort assignment problem that determines an assignment of parcels to primary sort equipment using an integer program, and (2) a secondary sort dispatch schedule problem that determines a dispatch schedule from primary to secondary sort areas for piles that require secondary sorting using a backward recursion algorithm. The assignment model is shown to preserve the feasibility of second-stage dispatch decisions while being computationally efficient and optimal for many practical objective functions, thereby enabling the generation of cost-effective plans that are time-feasible with respect

to parcel deadlines. However, the proposed model is deterministic and does not take into account the inherent stochasticity in demand, a missing practical consideration that is important to ensure the feasibility of sort operations, but requires more involved techniques.

To model demand uncertainty, different stochastic and robust optimization approaches have been proposed in the literature (see Bakker et al., 2020; Keith and Ahner, 2019; Sahinidis, 2004 for comprehensive reviews). In our work, a robust optimization framework is used to generate solutions that remain feasible for all possible demand realizations that belong to pre-defined uncertainty sets; a probability distribution is not required in this setting. Different robust optimization approaches have been proposed in the literature which vary mainly in their modeling choice as well as resulting solutions' level of conservatism. We refer the reader to Gabrel et al., 2014 and Bertsimas et al., 2011 for comprehensive reviews on the topic. In this paper, we use variations of the *budgeted* uncertainty models proposed by Bertsimas and Sim, 2004 to generate plans that are immunized against various changes in demand profiles. This type of uncertainty retains the complexity of the original problem. It also offers an explicit control on the level of conservatism of the resulting solutions using a budget of uncertainty, thereby allowing hub manager to evaluate trade-offs between operational costs and robustness levels to ultimately operate plans that are cost-effective under a desired level of protection against uncertainty.

3 Problem Description and Nominal Model

This paper considers a sort hub that operates a two-stage sort system with primary and secondary sorters designed to process small parcels and flats. A typical day in a sort hub is organized into non-overlapping sort shifts. Let \mathcal{T} denote the set of equally spaced time points that constitutes a sort shift, and suppose that a sort plan is to be designed for the shift.

The **primary sort stage** utilizes high-speed cross-belt sorters with a fixed number of discharge points that are mainly used for high-level grouping of parcels into *piles*. Define \mathcal{P} to be the set of primary sort piles that corresponds to these discharge points. Moreover, define $\mathcal{C} \subseteq \mathcal{T}$ to be the set of possible pile deadlines. A pile with deadline $c \in \mathcal{C}$ requires all of its assigned parcels to complete their sort requirements by time c . A pile may or may not require secondary sorting depending on the mix of parcels assigned to it. Let \mathcal{S} be the set of sort modes that indicates each pile's sort requirement, where $s = 1$ indicates that a pile is assigned a single destination and

service class and therefore does not require secondary sorting, while $s = 2$ indicates that a pile is assigned multiple destinations and/or service classes and requires secondary sorting.

The **secondary sort stage** operates multiple stations, each dedicated to processing one of the primary sort piles, with a processing rate of μ parcels per unit time (measured in the discretization of \mathcal{T}). A typical secondary sort station includes a fixed number of n sort positions (*e.g.*, cabinets or discharge points) depending on the configuration of the sort equipment used. Each station is responsible for processing a pre-assigned pile with secondary sort mode $s = 2$ by segregating its constituent parcels into one of the n sort positions. Parcels that complete their sort requirement are then packed into bags or other containers and moved to the loading area for shipping.

Each arriving parcel belongs to a *commodity* type; let \mathcal{K} be the set of all commodity types arriving to the sort hub during a sort shift. Each commodity $k \in \mathcal{K}$ is defined by a unique pair of a final destination and a sort time deadline d^k . All arriving parcels that belong to commodity k are required to finish their sorting no later than time d^k in order to meet their service guarantee for on-time delivery. Each commodity k is associated with a *nominal arrival profile* that represents its expected parcel arrival times $t \in \mathcal{T}$ and the associated expected arrival quantities q_t^k , measured in parcels. We assume that decisions related to parcel sequencing and loading onto primary sort belts are fixed, and therefore, each commodity's arrival profile correspond to parcels' arrival times to primary sort piles. Table 1 summarizes all of the notation used later in our modeling.

Table 1: Summary of sets and parameters

\mathcal{T}	set of integer time points t that constitutes a sort shift, $t \in \{1, 2, \dots, \mathcal{T} \}$
\mathcal{P}	set of primary sort piles $p \in \{1, 2, \dots, \mathcal{P} \}$
\mathcal{C}	set of deadlines, $\mathcal{C} \subseteq \mathcal{T}$
\mathcal{S}	set of secondary sort modes $s \in \{1, 2\}$
\mathcal{K}	set of commodities $k \in \{1, 2, \dots, \mathcal{K} \}$
μ	secondary sort processing rate, in parcels per unit time measured in the discretization of \mathcal{T}
d_k	deadline of commodity $k \in \mathcal{K}$, $d^k \in \mathcal{C}$
ℓ_k	expected last arrival of parcels with commodity type $k \in \mathcal{K}$, $\ell^l \in \mathcal{T}$
q_t^k	expected number of parcels with commodity type $k \in \mathcal{K}$ arriving at time $t \in \mathcal{T}$

For computational convenience, we introduce a pile-to-commodity compatibility rule and implement it in a pre-processing step to reduce model complexity. Specifically, we allow a commodity k to be assigned to a pile p with deadline c only if $d_k \geq c$ and $\ell_k \leq c - \epsilon$; that is, the commodity deadline is after the pile deadline and the commodity's expected last arrival time is ϵ

time units before the pile deadline, where ϵ is a user-defined slack value that should depend on specific hub operating conditions.

The goal of an operational planning model is to generate a *cost-effective* and *robust* assignment of commodities to primary sort pile positions while taking into account: (1) the limited capacity of secondary sorters, (2) parcel sort time deadlines, and (3) the inherent uncertainty in arrival quantities and/or arrival times of parcels that could have detrimental impact on sort plan performance as well as its time feasibility. In this section, we first present a nominal formulation addressing (1) and (2).

3.1 Nominal problem formulation

We recall an integer programming formulation for the *nominal primary-sort assignment (NPA)* problem that was first introduced in Khir et al., 2021 . The *NPA* determines a cost-effective assignment of commodities to piles using expected arrival quantities, denoted by q_t^k for every commodity k at time t , while taking into account limited sort capacities as well as parcel deadlines.

The decision variables are:

$$y_{p,c,s} = \begin{cases} 1, & \text{if pile } p \text{ with deadline } c \text{ and sort mode } s \text{ is used} \\ 0, & \text{otherwise,} \end{cases}$$

and

$$x_{p,c,s}^k = \begin{cases} 1, & \text{if commodity } k \text{ is assigned to pile } p \text{ with deadline } c \text{ and sort mode } s \\ 0, & \text{otherwise.} \end{cases}$$

The *NPA* integer programming model is formulated as follows:

$$\text{Minimize}_{x,y} \quad c(x,y) \tag{1a}$$

$$\text{subject to} \quad \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} x_{p,c,s}^k = 1 \quad \forall k \in \mathcal{K}, \tag{1b}$$

$$\sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} y_{p,c,s} \leq |\mathcal{P}|, \tag{1c}$$

$$\sum_{k \in \mathcal{K}} x_{p,c,2}^k \leq ny_{p,c,2} \quad \forall p \in \mathcal{P}, c \in \mathcal{C}, \tag{1d}$$

$$\sum_{k \in \mathcal{K}} x_{p,c,1}^k \leq y_{p,c,1} \quad \forall p \in \mathcal{P}, c \in \mathcal{C}, \tag{1e}$$

$$\sum_{k \in \mathcal{K}} \sum_{t < \tau \leq c} q_{\tau}^k x_{p,c,2}^k \leq \mu(c - t) \quad \forall t \in \mathcal{T}, p \in \mathcal{P}, c \in \mathcal{C}. \quad (1f)$$

$$x_{p,c,s}^k \in \{0, 1\} \quad \forall k \in \mathcal{K}, p \in \mathcal{P}, c \in \mathcal{C}, \forall s \in \mathcal{S}, \quad (1g)$$

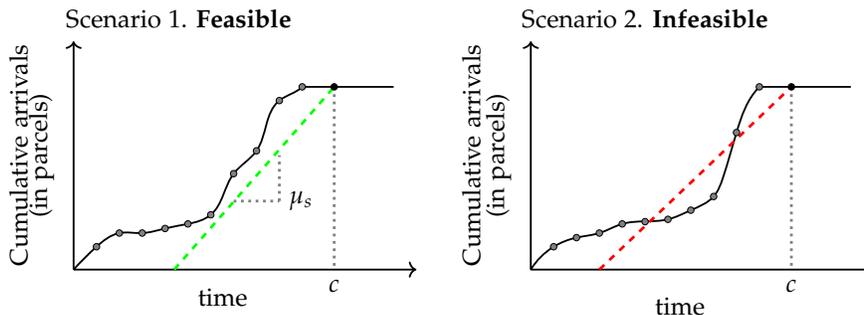
$$y_{p,c,s} \in \{0, 1\} \quad \forall p \in \mathcal{P}, c \in \mathcal{C}, \forall s \in \mathcal{S}. \quad (1h)$$

The objective function (1a) is written in a generic form $c(x, y)$, and is assumed to be some linear function of these decision variables. Later in Section 5, we describe possible objective functions; for the purpose of our computational experiments, we model two objectives that we derived in consultation with our industry partner and optimize for them in lexicographical fashion.

Constraints (1b) ensure that each commodity is assigned to only one primary sort pile, *i.e.*, its associated parcels cannot be split across multiple piles. Constraints (1c)-(1d) are capacity-related constraints that are dependent on the configuration of sort equipment employed at each stage. Constraint (1c) ensures that no more than $|\mathcal{P}|$ primary sort piles are used, where $|\mathcal{P}|$ is the number of available discharge points in primary sort belts. Similarly, constraints (1d) ensure that no more than n commodities are assigned to primary sort piles with sort mode $s = 2$, where n is the number of available sort positions in each secondary sort station. A primary sort pile with sort mode $s = 1$ does not require secondary sorting and, therefore, is allowed to be assigned at most one commodity, as modeled in constraints (1e). Finally, constraints (1f) are time-feasibility constraints to ensure that all commodities that require secondary sorting meet their sort time deadlines. The intuition behind these constraints is based on simple deterministic queuing analysis using cumulative arrival and service curves, as illustrated in Figure 2 and explained in more detail in Khir et al., 2021. In particular, these constraints guarantee that each point on the cumulative arrival curve for every pile lies above its secondary sort deadline line, which has a slope equal to the secondary sort rate μ ; this consequently ensures that there is enough secondary sort capacity to process all arrivals to every secondary sort pile by their deadline. Note that constraint set (1f) in its current form presents a special case where the time it takes to carry out non-sort related activities is negligible. That is, activities like transfers between primary and secondary sort areas or equipment configuration do not result in noticeable loss in sort capacity. The model, however, can be easily adjusted to account for such losses. More specifically, the right-hand side of constraints (1f) can be changed to $\mu(c - t - r)$; this corresponds to shifting the secondary-sort deadline line in Figure 2 by r time units to the left, where r is measured in the discretization of \mathcal{T} , thereby

reducing the remaining available secondary sort time and capacity.

Figure 2: An example to illustrate the modeling idea behind constraints (1f) to ensure time-feasibility of pile assignments. Under constraint (1f), scenario 1 is feasible because every point on the arrival curve for a pile (grey curve) lies above its secondary-sort deadline line (green dashed), while scenario 2 is infeasible, meaning there is not enough capacity to process all assigned arrivals by the pile’s deadline.



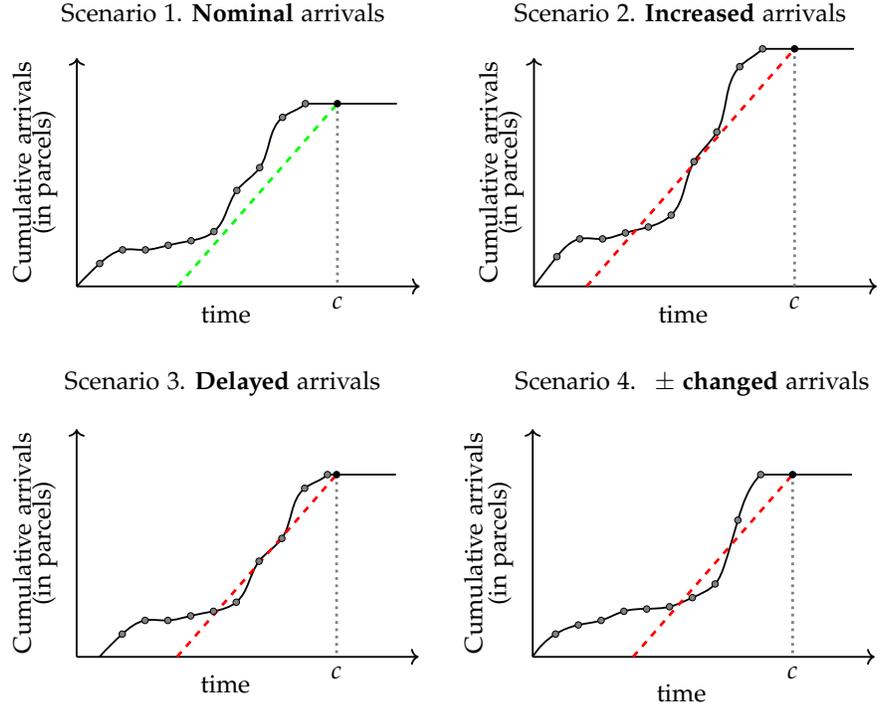
4 Robust Problem Formulations

The *NPA* model is guaranteed to generate feasible sort operations as long as the *actual* arrival profiles for a sort shift are precisely known and used when generating sort plans. However, in practice, hub managers typically generate sort plans in advance of operations using *expected* arrival profiles, and fix them for several weeks to achieve stability and efficiency in operations. It is not difficult to see then that variations from these expected arrival profiles could lead to infeasibility during an actual operational shift; specifically, they could lead to violations of constraints (1f).

To illustrate, consider a pile assignment generated using the *NPA* formulation as shown in Figure 3. Scenario 1 shows a feasible pile assignment using nominal arrival profiles. In an actual demand realization, parcel arrival quantities could increase from their nominal arrival quantities (scenario 2), parcels could arrive later than their expected arrival times (scenario 3), and parcel quantities could simply increase or decrease over time (scenario 4). Clearly, these changes could lead to infeasibility, as sorters are no longer guaranteed to have enough capacity to process all arrivals by their deadline.

To address demand uncertainty, we develop a robust optimization approach to generate plans that are protected against different types of uncertainty commonly experienced in practice, as discussed in the preceding examples. In particular, we seek to generate pile assignments that remain feasible for all possible demand realizations that belong to pre-defined uncertainty sets.

Figure 3: An example of a nominal pile assignment under different demand realizations



We use the interval budgeted uncertainty representation proposed in Bertsimas and Sim, 2004 to encode our knowledge of possible demand realizations. The size of these sets is controlled by an uncertainty budget to allow planners to gain insights about trade-offs between operational costs and robustness, to ultimately operate plans that are cost-effective and reflective of their desired level of risk tolerance.

Recall that we use q_t^k to represent nominal arrival quantities for every commodity k at time t . We now introduce \tilde{q}^k to denote a potential demand realization for commodity k , where $\tilde{q}^k = (\tilde{q}_0^k, \tilde{q}_1^k, \dots, \tilde{q}_{|\mathcal{T}|}^k)$. The goal is to design plans that are guaranteed to remain feasible for all demand realizations \tilde{q}^k that belong to an uncertainty set \mathcal{Q} . We consider the following three types of uncertainty sets, illustrated in Figure 4:

Commodity-based model: Every commodity $k \in \mathcal{K}$ can have its arrival profile scaled up by a fraction $\alpha \in [0, 1]$ in the worst case; the parcel volume arriving then by any time t may be $1 + \alpha$ times the nominal projection at that same time. To control conservatism, changes from nominal forecast values are limited to Π commodities in total. Therefore, we define

$$\mathcal{Q}_{\text{commodity}} = \left\{ \tilde{q} : \tilde{q}^k = q^k + \alpha q^k \zeta^k, \quad \sum_{k \in \mathcal{K}} \zeta^k \leq \Pi, \quad 0 \leq \zeta^k \leq 1, \quad \forall k \in \mathcal{K} \right\}$$

where ζ^k is an independent random variable subject to uncertainty.

Parcel-based model: Every commodity $k \in \mathcal{K}$ can have its arrival profile scaled up or down by a fraction $\alpha \in [0, 1]$ in the worst case. To control conservatism, *total absolute changes* in arrivals from nominal forecast values are limited to Γ parcels in total across all commodities throughout a sort shift, and total arrivals are not allowed to exceed the total nominal arrival quantities throughout a sort shift by more than a fraction $\Delta \in [0, 1]$. Therefore, we define

$$\mathcal{Q}_{\text{parcel}} = \left\{ \tilde{q} : \tilde{q}^k = q^k + \alpha q^k \zeta^k, \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} \alpha q^k |\zeta^k| \leq \Gamma, \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} \alpha q^k \zeta^k \leq \Delta \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} q^k, -1 \leq \zeta^k \leq 1, \forall k \in \mathcal{K} \right\}.$$

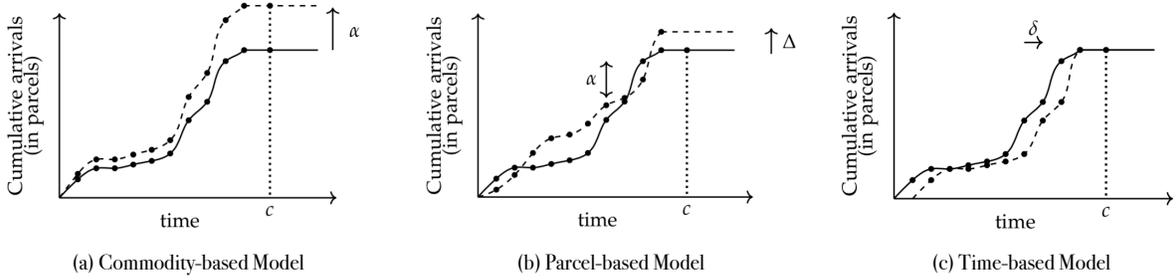
Time-based model: Parcels can be delayed to arrive after their expected arrival time by at most δ time units measured in the discretization of \mathcal{T} , in the worst case, as long as they arrive by their latest feasible arrival time ℓ^k . Delays in arrivals are limited to Γ parcels in total across all commodities. Therefore, we define

$$\mathcal{Q}_{\text{time}} = \left\{ \tilde{q} : \tilde{q}_t^k = q_t^k + \sum_{t-\delta \leq \tau < t} q_\tau^k \zeta_\tau^k, \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} q_\tau^k \zeta_\tau^k \leq \Gamma, 0 \leq \zeta_t^k \leq 1, \forall k \in \mathcal{K}, t \in \mathcal{T}, t \leq \ell^k \right\}.$$

We limit our presentation and computational experiments to these three primary models of demand uncertainty that separately capture changes in either arrival quantities or arrival times. While these changes are typically experienced simultaneously in the real world, it is likely that one source of uncertainty is more significant than the others, depending on characteristics of the particular sorting hub under consideration (hub location, size, etc.). Therefore, treating each uncertainty source independently provides hub managers with a better understanding of the impact each has on operational performance measures, ultimately directing attention to design options that better suit hub-specific dynamics. We note that it is also possible to build robust model variants that capture changes in both parcel arrival volumes and times, by combining our modeling ideas; a specific example that allows for parcel increases, decreases, and/or delays is presented in the Appendix.

In the remainder of the section, we discuss each of the uncertainty models in more detail and formulate the corresponding robust counterparts of *NPA*. Notice that constraint set (1f) is the only set of constraints that requires modification, since it is the only set of constraints that involves the uncertain demand quantity \tilde{q}_t^k . Therefore, we reformulate constraints (1f) to generate assignments that are immunized against the different sources of uncertainty described earlier. The general robust reformulation is

Figure 4: An illustration of the three models of uncertainty considered, each discussed in more detail in the following subsections.



$$\begin{aligned}
 & \underset{x,y}{\text{Minimize}} && c(x,y) && (2a) \\
 & \text{subject to} && (1b) - (1e) && (2b) - (2e) \\
 & && \sum_{k \in K} \sum_{t < \tau \leq c} q_{\tau}^k x_{p,c,2}^k + \beta_{t,p,c,2}(B, \bar{x}) \leq \mu(c-t) && \forall t \in \mathcal{T}, p \in \mathcal{P}, c \in \mathcal{C}. \quad (2f)
 \end{aligned}$$

Here, $\beta_{t,p,c,2}(B, \bar{x})$ is a protection function, a buffer that we quantify based on our choice of the uncertainty set and the desired level of robustness B , which is represented by Π in the commodity-based model and Γ in the parcel- and time-based models.

4.1 Commodity-based formulation

In this robust model, the goal is to generate sort plans that are protected against *increases* in commodity demand arriving during a sort shift, which could adversely impact the operational performance of sort plans and their time feasibility. Suppose that \tilde{q}^k , the uncertain demand of commodity k , takes a random value in the interval $[q^k, (1 + \alpha)q^k]$, where α is a maximum possible deviation from a nominal demand value. In reality, it is unlikely that all commodities' demand will change to their worst value simultaneously. We denote Π to be the uncertainty budget, measured in commodities, which we assume to be an integer here. Our goal is to generate plans that are protected against all cases with up to Π commodities having their arrival profiles scaled up by α in the worst case. Notice that in this model of uncertainty, we consider only positive scaled deviation from commodities' arrival profiles, as shown in Figure 4(a).

In order to generate a robust counterpart, we first define an adversary problem corresponding

to this model of uncertainty. Given an assignment vector \bar{x} , the protection function $\beta_{t,p,c,2}(\Pi, \bar{x})$ is given by the optimum of the following optimization problem:

$$\beta_{t,p,c,2}(\Pi, \bar{x}) = \underset{\lambda}{\text{Maximize}} \quad \sum_{k \in \mathcal{K}} \sum_{t < \tau \leq c} [\alpha q_{\tau}^k \bar{x}_{p,c,2}^k] \lambda^k \quad (3a)$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \lambda^k \leq \Pi, \quad (3b)$$

$$0 \leq \lambda^k \leq 1 \quad \forall k \in \mathcal{K}. \quad (3c)$$

The objective function (3a) measures the number of parcels that could arrive to pile p with deadline c during the interval $(t, c]$ on top of its assigned nominal demand, which is to be maximized from an adversarial perspective. Constraint (3b) sets a limit on the number of commodities that could contribute to such an increase in arrival quantities by allowing up to Π commodities to assume their worst-case deviation, where Π here is a user-defined input parameter that is used to control the desired level of conservatism of the generated plans. Since the constraint matrix of this linear program is totally unimodular, at an optimal extreme point solution the variable λ^k equals 1 if commodity k assumes its worst-case deviation during the sort shift, and 0 otherwise. We next present the corresponding robust formulation for the commodity-based uncertainty model.

Proposition 1. *The robust counterpart of the NPA formulation under the commodity-based uncertainty model can be formulated as the following mixed-integer program:*

$$\underset{x,y,z,u}{\text{Minimize}} \quad c(x, y) \quad (4a)$$

$$\text{subject to} \quad (1b) - (1e) \quad (4b) - (4e)$$

$$\sum_{k \in \mathcal{K}} \sum_{t < \tau \leq c} q_{\tau}^k x_{p,c,2}^k + \Pi z_{t,p,c,2} + \sum_{k \in \mathcal{K}} u_{t,p,c,2}^k \leq \mu(c - t) \quad \forall t \in \mathcal{T}, p \in \mathcal{P}, c \in \mathcal{C}, \quad (4f)$$

$$z_{t,p,c,2} + u_{t,p,c,2}^k \geq \sum_{t < \tau \leq c} \alpha q_{\tau}^k x_{p,c,2}^k \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}, c \in \mathcal{C}. \quad (4g)$$

Proof. We first consider the following dual formulation of the commodity-based adversary problem, modeled in (3):

$$\underset{z,u}{\text{Minimize}} \quad \Pi z + \sum_{k \in \mathcal{K}} u^k \quad (5a)$$

$$\text{subject to} \quad z + u^k \geq \sum_{t < \tau \leq c} \alpha q_{\tau}^k \bar{x}_{p,c,2}^k \quad \forall k \in \mathcal{K}, \quad (5b)$$

$$z \geq 0, \quad (5c)$$

$$u^k \geq 0 \quad \forall k \in \mathcal{K}. \quad (5d)$$

Since Model (3) is feasible and bounded, its dual formulation (5) is also feasible and bounded by strong duality, and therefore, they generate the same objective function value. It follows then that the protection function $\beta_{t,p,c,2}(\Pi, \bar{x})$ equals the optimum of the dual formulation (5). Substituting (5) in Model (2), we obtain the robust mixed-integer formulation (4) under the commodity-based uncertainty set. \square

Constraints (4f)-(4g) in Model (4) replace constraints (1f) in the nominal model *NPA*. While the resulting reformulation (4) requires adding up to $(|\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|) + (|\mathcal{K}| \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|)$ continuous variables and up to $(|\mathcal{K}| \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|)$ constraints, it is still tractable for realistic-sized instances found in practice as we show later in Section 5. Notice that the implicit maximum value of the uncertainty budget Π under the commodity-based model equals $\min(|\mathcal{K}|, n)$, where $|\mathcal{K}|$ is the total number of commodities arriving during a sort shift and n is the number of commodities that can be assigned to the same pile with secondary sort mode $s = 2$. Therefore, $\Pi = \min(n, |\mathcal{K}|)$ corresponds to the maximum protection case under the commodity-based model, while $\Pi = 0$ corresponds to the nominal case.

4.2 Parcel-based formulation

In this robust model, the goal is to generate sort plans that are protected against *increases or decreases* in commodity demand arriving during a sort shift. That is, we now assume that $\tilde{q}^k \in [(1 - \alpha)q^k, (1 + \alpha)q^k]$, where α is still defined as a pre-specified deviation from nominal demand values. Unlike the commodity-based formulation presented earlier, in this model we express the uncertainty budget in terms of parcel count and not commodity count, i.e., we limit the number of positive or negative changes from nominal demands to at most Γ parcels. Furthermore, we place an additional restriction on aggregate changes throughout the sort shift. In particular, we limit increases in total arrivals to a fraction $\Delta \in [0, 1]$ from the nominal total arrival quantities. For instance, if the expected total arrivals during a sort shift is 10,000 parcels, and $\Delta = 0.1$, then we want to consider cases that only result in at most 11,000 parcels arriving at the hub in total across all commodities. This limitation reflects practical insights suggesting it is more common to have accurate forecasting for aggregate arrival quantities during a shift rather than than accurate commodity-specific arrival patterns. Figure 4(b) shows an example of a cumulative arrival profile admissible under this model of uncertainty.

Similar to our derivation of the commodity-based formulation, we first define an adversary problem corresponding to the parcel-based uncertainty model described earlier. Given an assignment vector \bar{x} , the protection function $\beta_{t,p,c,2}(\Gamma, \bar{x})$ equals the optimum of the following optimization problem:

$$\beta_{t,p,c,2}(\Gamma, \bar{x}) = \underset{\lambda, \pi}{\text{Maximize}} \quad \sum_{k \in K} \sum_{t < \tau \leq c} \alpha q_{\tau}^k \bar{x}_{p,c,2}^k (\lambda^k - \pi^k) \quad (6a)$$

$$\text{subject to} \quad \sum_{k \in K} \sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (\lambda^k + \pi^k) \leq \Gamma, \quad (6b)$$

$$\sum_{k \in K} \sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (\lambda^k - \pi^k) \leq \Delta \left(\sum_{k \in K} \sum_{\tau \in \mathcal{T}} q_{\tau}^k \right), \quad (6c)$$

$$0 \leq \lambda^k \leq 1 \quad \forall k \quad (6d)$$

$$0 \leq \pi^k \leq 1 \quad \forall k \quad (6e)$$

The objective function (6a) maximizes the changes, positive and negative, from nominal arrival quantities for all commodities assigned to pile p with deadline c during the interval $(t, c]$. Constraint (6b) limits these changes to Γ parcels in total. Constraint (6c) ensures that the total increases in arrival quantities do not exceed the total nominal arrivals by more than a fraction $\Delta \in [0, 1]$. In proposition 2, we use this adversary setting to derive a robust formulation that corresponds to the parcel-based uncertainty model.

Proposition 2. *The robust counterpart of the NPA formulation under the parcel-based uncertainty model can be formulated as the following mixed-integer program:*

$$\underset{x,y,z,s,u,w}{\text{Minimize}} \quad c(x, y) \quad (7a)$$

$$\text{subject to} \quad (1b) - (1e) \quad (7b) - (7e)$$

$$\sum_{k \in K} \sum_{t < \tau \leq c} q_{\tau}^k x_{p,c,2}^k + \Gamma z_{t,p,c,2} + \sum_{k \in K} \left(\Delta \sum_{\tau \in \mathcal{T}} q_{\tau}^k s_{t,p,c,2} + u_{t,p,c,2}^k + w_{t,p,c,2}^k \right) \leq \mu(c - t) \quad \forall t, p, c, \quad (7f)$$

$$\sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (z_{t,p,c,2} + s_{t,p,c,2}) + u_{t,p,c,2}^k \geq \sum_{t < \tau \leq c} \alpha q_{\tau}^k x_{p,c,2}^k \quad \forall k, t, p, c, \quad (7g)$$

$$\sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (z_{t,p,c,2} - s_{t,p,c,2}) + w_{t,p,c,2}^k \geq - \sum_{t < \tau \leq c} \alpha q_{\tau}^k x_{p,c,2}^k \quad \forall k, t, p, c. \quad (7h)$$

Proof. We first consider the following dual formulation of the parcel-based adversary problem, modeled in (6):

$$\underset{z,s,w,u}{\text{Minimize}} \quad \Gamma z + \sum_{k \in K} \left(\Delta \sum_{\tau \in \mathcal{T}} q_{\tau}^k s + u^k + w^k \right) \quad (8a)$$

$$\text{subject to} \quad \sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (z + s) + u^k \geq \sum_{t < \tau \leq c} \alpha q_{\tau}^k \bar{x}_{p,c}^k \quad \forall k \quad (8b)$$

$$\sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (z - s) + w^k \geq - \sum_{t < \tau \leq c} \alpha q_{\tau}^k \bar{x}_{p,c,s}^k \quad \forall k \quad (8c)$$

$$z, s, \geq 0 \tag{8d}$$

$$w^k, u^k \geq 0 \quad \forall k. \tag{8e}$$

Since Model (6) is feasible and bounded, its dual formulation (8) is also feasible and bounded by strong duality, and therefore, they generate the same objective function value. It follows then that the protection function $\beta_{t,p,c,2}(\Gamma, \bar{x})$ equals the optimum of the dual formulation (8). Substituting (8) in Model (2), we obtain the robust mixed-integer formulation (7) under the parcel-based uncertainty set. \square

Constraints (7f)-(7h) replace constraints (1f) in the nominal model *NPA*. The resulting reformulation (7) requires adding up to $(2 \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|) + (2 \times |\mathcal{K}| \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|)$ continuous variables and up to $(2 \times |\mathcal{K}| \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|)$ constraints. The resulting model is still tractable for realistic-sized instances found in practice as we show later in Section 5.

4.3 Time-based formulation

In this robust model, the goal is to generate sort plans that are protected against *delays* in demand arrivals. We consider cases where parcels can be delayed to arrive after their expected arrival time by δ time units in the worst case, as long as they arrive by their latest feasible arrival time ℓ^k , $\bar{q}_t^k \in [q_t^k, \sum_{t-\delta \leq \tau \leq t} q_\tau^k]$ for $t \leq \ell^k$. As in the previous model, we limit the number of changes to Γ parcels, i.e., we consider cases where at most Γ parcels could arrive later than their expected arrival times. Note that this model of uncertainty considers changes only in arrival times and not in arrival quantities, as illustrated in Figure 4(c).

Similar to our derivation of the commodity- and parcel-based formulations, we first define an adversary problem corresponding to the time-based uncertainty model described earlier. Given an assignment vector \bar{x} , the protection function $\beta_{t,p,c,2}(\Gamma, \bar{x})$ equals the optimum of the following optimization problem:

$$\beta_{t,p,c,2}(\Gamma, \bar{x}) = \underset{\lambda}{\text{Maximize}} \quad \sum_{k \in \mathcal{K}: \ell^k > t} \sum_{t-\delta+1 \leq \tau \leq t} [q_\tau^k \bar{x}_{p,c,2}^k] \lambda_\tau^k \tag{9a}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} q_\tau^k \lambda_\tau^k \leq \Gamma, \tag{9b}$$

$$0 \leq \lambda_\tau^k \leq 1 \quad \forall k \in \mathcal{K}, \tau \in \mathcal{T}. \tag{9c}$$

The objective function (9a) maximizes the number of parcels that could arrive due to delays to pile p with deadline c during the interval $(t, c]$. Constraint (9b) limits the number of parcels that

could arrive due to delays to at most Γ parcels. Proposition 3 presents the robust formulation corresponding to this model of the adversary problem.

Proposition 3. *The robust counterpart of the NPA formulation under the time-based uncertainty model can be formulated as the following mixed-integer program:*

$$\underset{x,y,z,u}{\text{Minimize}} \quad c(x,y) \tag{10a}$$

$$\text{subject to} \quad (1b) - (1e) \tag{10b) - (10e)}$$

$$\sum_{k \in \mathcal{K}} \sum_{t < \tau \leq c} q_{\tau}^k x_{p,c,2}^k + \Gamma z_{t,p,c,2} + \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} u_{\tau,t,p,c,2}^k \leq \mu(c-t) \quad \forall t,p,c, \tag{10f}$$

$$q_{\tau}^k z_{t,p,c,2} + u_{\tau,t,p,c,2}^k \geq q_{\tau}^k \bar{x}_{p,c,2}^k \quad \forall p,c,k,t,\tau, \quad \ell^k > t, t - \delta + 1 \leq \tau \leq t. \tag{10g}$$

Proof. We first consider the following dual formulation of the time-based adversary problem, modeled in (9):

$$\underset{z,u}{\text{Minimize}} \quad \Gamma z + \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} u_{\tau}^k \tag{11a}$$

$$\text{subject to} \quad q_{\tau}^k z + u_{\tau}^k \geq q_{\tau}^k \bar{x}_{p,c,2}^k \quad \forall k, \tau, \quad \ell^k > t, \quad t - \delta + 1 \leq \tau \leq t, \tag{11b}$$

$$z \geq 0, \tag{11c}$$

$$u_{\tau}^k \geq 0 \quad \forall k \in \mathcal{K}, \tau \in \mathcal{T}. \tag{11d}$$

Since Model (9) is feasible and bounded, its dual formulation (11) is also feasible and bounded by strong duality, and therefore, they generate the same objective function value. It follows then that the protection function $\beta_{t,p,c,2}(\Gamma, \bar{x})$ equals also to the objective function of the dual formulation (11). Substituting (11) in Model (2), we obtain the robust mixed-integer formulation (10) under the time-based uncertainty set. \square

Notice that the size of this formulation depends heavily on the choice of the time discretization used in the model, which needs to be reasonably fine (e.g. 30 minutes) to capture the time-sensitive nature of sort operations. Specifically, it requires adding up to $(|\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{K}| \times |\mathcal{T}|^2)$ constraints and up to $(|\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|^2) + (|\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{K}| \times |\mathcal{T}|^2)$ variables to the nominal formulation, which could pose a computational challenge when solving a realistic sized-instance. To circumvent this issue, we consider a reformulation for the time-based uncertainty model using the the following observation.

Proposition 4. *An optimal solution for (9) has an objective function value equal to*

$$\min \left(\Gamma, \sum_{k \in \mathcal{K}: \ell^k > t} \sum_{t-\delta+1 \leq \tau \leq t} q_\tau^k \bar{x}_{p,c,2}^k \right).$$

Proof. Recall that for a given budget Γ and an assignment vector \bar{x} , the adversary problem modeled in (9) is solved for every pile p with deadline c at time t . Clearly, a commodity that is not assigned to pile p with deadline c has no impact on its objective function value in Model (9), and thus, an optimal solution for Model (9) has $\lambda_\tau^k = 0$ for all commodity k such that $\bar{x}_{p,c,2}^k = 0$. Additionally, among all commodities with $\bar{x}_{p,c,2}^k = 1$, only those that are eligible for delays, i.e., parcels that belong to commodities with $\ell^k > t$, could impact the objective function value. Then, it follows from constraint (9b) that the maximum value of the objective function (9a) is either Γ , the uncertainty budget, or the total number of parcels that are assigned to the pile and expected to arrive in the past δ periods with $\ell^k > t$, whichever is smaller. \square

Using Proposition 4, the adversary problem can be rewritten as

$$\text{Maximize}_{\lambda} \quad \sum_{k \in \mathcal{K}} \bar{x}_{p,c,2}^k \lambda^k \quad (12a)$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \lambda^k \leq \Gamma, \quad (12b)$$

$$0 \leq \lambda^k \leq \sum_{t-\delta+1 \leq \tau \leq t} q_\tau^k \quad \forall k \in \mathcal{K} : \ell^k > t. \quad (12c)$$

The objective function (12a) maximizes the number of parcels that could arrive due to delays to pile p with deadline c during the interval $(t, c]$. Constraint (12b) limits the number of parcels that could arrive due to delays to at most Γ parcels. In Proposition 5, we use this version of the adversary problem to specify the corresponding robust formulation for the time-based uncertainty model.

Proposition 5. *The robust counterpart of the NPA formulation under the time-based uncertainty model can be formulated as the following mixed-integer program:*

$$\text{Minimize}_{x,y,z,\mu} \quad c(x, y) \quad (13a)$$

$$\text{subject to} \quad (1b) - (1e) \quad (13b) - (13e)$$

$$\sum_{k \in \mathcal{K}} \sum_{t < \tau \leq c} q_\tau^k x_{p,c,2}^k + \Gamma z_{t,p,c,2} + \sum_{k \in \mathcal{K}: \ell^k > t} \sum_{t-\delta+1 \leq \tau \leq t} q_\tau^k u_{t,p,c,2}^k \leq \mu(c-t) \quad \forall t, p, c, \quad (13f)$$

$$z_{t,p,c,2} + u_{t,p,c,2}^k \geq x_{p,c,2}^k \quad \forall t, p, c, k, \ell^k > t. \quad (13g)$$

Proof. We consider the dual formulation of the time-based adversary problem given in (12):

$$\text{Minimize}_{z,u} \quad \Gamma z + \sum_{k \in \mathcal{K}: \ell^k > t} \sum_{t-\delta+1 \leq \tau \leq t} q_\tau^k u^k \quad (14a)$$

$$\text{subject to} \quad z + u^k \geq x_{p,c,2}^k \quad \forall k \in \mathcal{K} : \ell^k > t, \quad (14b)$$

$$z \geq 0, \quad (14c)$$

$$u^k \geq 0 \quad \forall k \in \mathcal{K}. \quad (14d)$$

Since (12) is feasible and bounded, its dual formulation (14) is also feasible and bounded by strong duality, and therefore, they generate the same objective function value. It follows that the protection function $\beta_{t,p,c,2}(\Gamma, \bar{x})$ equals the optimum of the dual formulation (14). Substituting (14) in (2), we obtain the robust mixed-integer formulation (13) under the time-based uncertainty set. \square

Constraints (13f)-(13g) replace constraints (1f) in the nominal model *NPA*. The resulting reformulation (13) requires adding up to $(|\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|) + (|\mathcal{K}| \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|)$ continuous variables and up to $(|\mathcal{K}| \times |\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{T}|)$ constraints. It is less dependent on the choice of time discretization in the model when compared to (10), and is computationally efficient for solving realistic-sized instances, as we show in Section 5.

5 Computational Study

The primary objectives of our experiments are to: (1) assess the computational performance of the proposed robust models in terms of efficiency and solution quality, and (2) illustrate their value by analyzing trade-offs between operational costs and various levels of robustness. Motivated by our models' computational tractability, we also investigate a model extension that allows for more flexibility in sort operations; specifically, we allow pile assignments to be reconfigured once during a sort shift. The model extension's goal is to test whether introducing limited flexibility to sort operations can help reduce the *price of robustness* of the sort plans while protecting against a specific level of uncertainty.

We consider test instances based on data obtained from a major international parcel carrier. When solving the models, we use a hierarchical optimization framework that optimizes for two

objectives: First, we minimize the expected number of required secondary sort hours, modeled as

$$\text{Minimize } \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} cy_{p,c,2}.$$

We then fix the shift’s total secondary sort hours and optimize for a second objective, which is to minimize the number of parcels that require secondary sorting (or equivalently, maximizing the expected number of parcels that are directly sorted using primary sort equipment), to maximize the utilization of primary sort equipment by reducing workload assigned to secondary sorters, modeled as

$$\text{Minimize } \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} q_t^k x_{p,c,2}^k.$$

Recall that in a two-stage sort system, each arriving parcel is first sorted in the primary stage. If the parcel is assigned to a pile with secondary mode $s = 2$, then it must also go to secondary sorting to complete its sort requirement. Reducing the number of hours spent on the secondary sort of parcels reduces operational costs, including for example the power and other operating costs associated with operating fully-automated secondary sorters. While we use this objective function in this paper, the solution framework proposed is quite flexible and can accommodate other potential objectives. Alternative objective functions are discussed later in Section 5.4.2.

All models were implemented in Python 3.8, and solved using Gurobi 9.0.2. A Linux computing cluster, which employs HTCondor version 8.8.9 and machines with 8 GB memory, was used to carry out the computations.

5.1 Description of test instances

We consider a sort system with $|\mathcal{P}| = 16$ primary sort pile positions, $n = 15$ secondary sort positions, and a secondary sort rate μ of 800 parcels per hour per station. We model all operations using a 30-minute time discretization, i.e., each time period t is 30 minutes long. We use demand data obtained from a large parcel carrier as basis for generating instances of different size and composition. The data consist of parcel flow information during different sort shifts with peak arrivals taking place during night hours. A general description of our test instances, including operational shift structure and demand volume, is given in Table 2.

Table 2: Characteristics of test instances created using data obtained from an international service provider

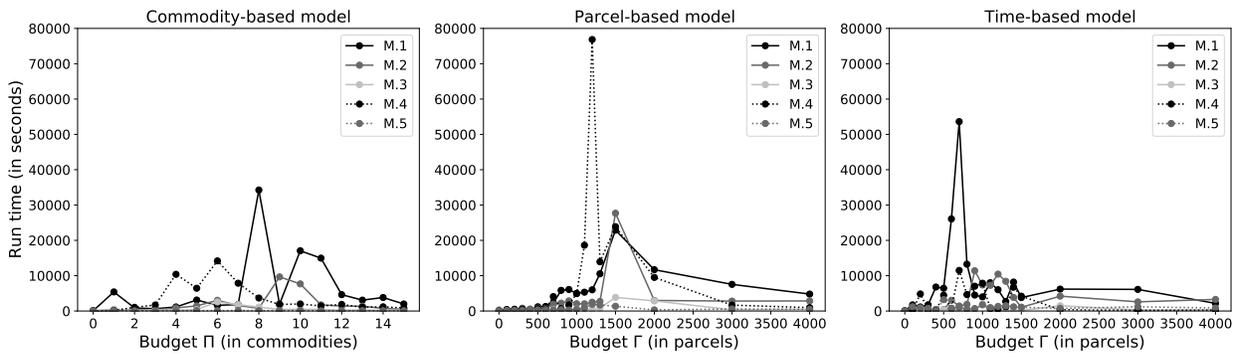
Sort Shift	Instance	Total arrivals parcels	Total arrivals commodities	Average # of parcels/commodity	Variability between commodity sizes (CoV)
Daysort (12pm - 4pm)	D.1	5682	45	126	0.80
	D.2	4524	41	110	0.69
Twilight (4pm - 9pm)	T.1	7104	35	203	0.72
	T.2	8399	39	215	0.63
	T.3	10183	42	243	0.70
Midnight (9pm - 4am)	M.1	25470	38	670	0.71
	M.2	28654	40	716	0.78
	M.3	30082	35	860	0.84
	M.4	31892	40	797	0.82
	M.5	28238	32	883	0.73

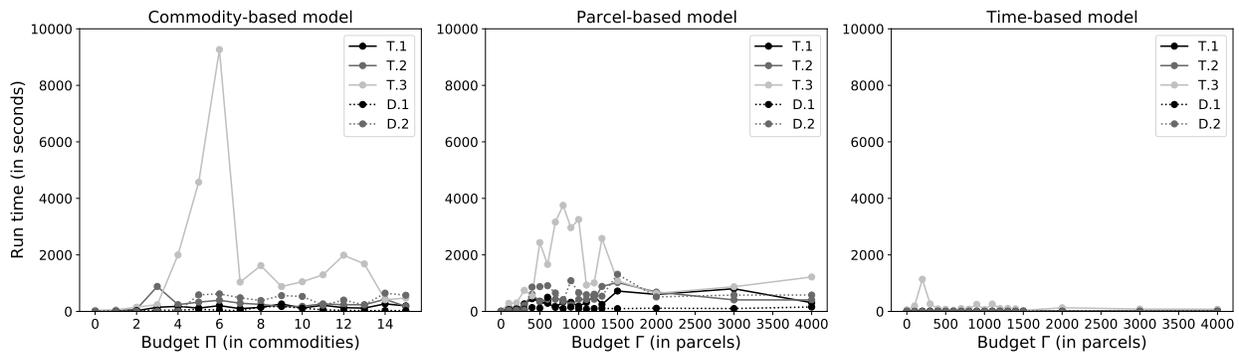
We consider cases where demand can deviate by at most 30% from nominal values, i.e., we set $\alpha = 0.3$ in the commodity- and parcel-based models. We also allow demand to be delayed by at most 2 periods from expected arrival times, i.e., we set $\delta = 2$, which corresponds to a maximum of 60 minutes delay in the time-based model. In the parcel-based model, we consider cases where the total arrivals during a sort shift won't deviate from the expected total arrivals during a given sort shift, i.e., we set $\Delta = 0$.

5.2 Computational performance

We first evaluate the computational performance of each of the robust models in terms of run time and solution quality. Figure 5 summarizes the run times for each of the instances over different values of the uncertainty budgets, when solved to optimality using the three robust formulations.

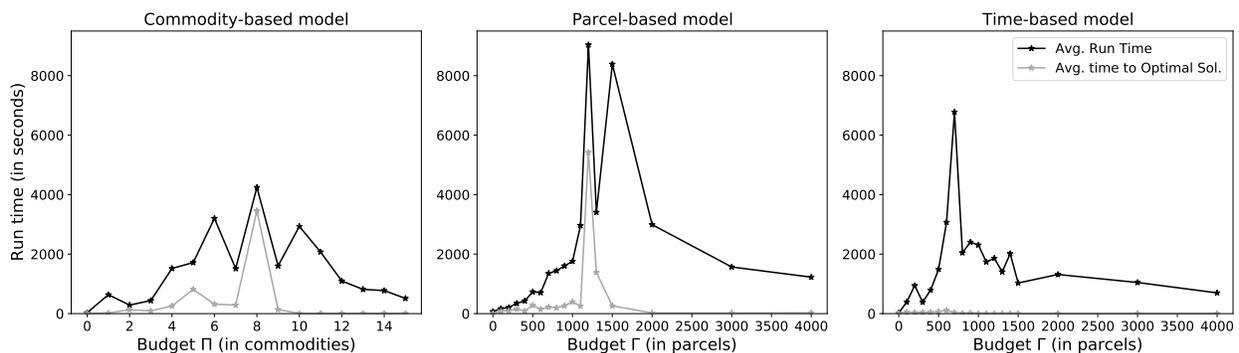
Figure 5: Run time to optimality for various robustness levels





As the figures indicate, in all the three models, optimal solutions are found in relatively short time for the nominal and maximum protection cases, i.e., when the uncertainty budget equals to zero and when it is large enough to allow for all commodities/parcels to change to their worst-case values. It takes relatively longer to solve instances when the robustness level is in the middle part of the range. While the average time to solve the commodity-, parcel-, and time-based models to proven optimality is 1462 seconds, 2133 seconds, and 1670 seconds, respectively, it takes on average about 347 seconds, 502 seconds, and 22 seconds, respectively, to actually find an optimal solution. Figure 6 summarizes the time it takes to find an optimal solution versus the time it takes to find it and prove its optimality, averaged over all test instances, for various robustness levels. This shows that high-quality solutions can be generated in relatively short time (under two hours), which is practical given that sort assignments are generated every few weeks.

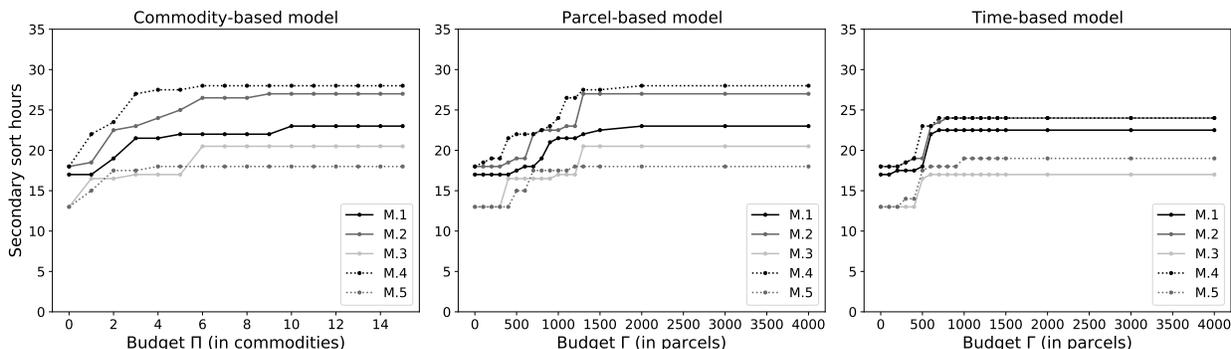
Figure 6: Time taken to solve instances to proven optimality compared to time taken to find an optimal solution, averaged over all test instances



5.3 Operational insights

We take a closer look into the solutions generated using the three robust formulations. We focus on Midnight shift instances, the largest and busiest sort shifts, and therefore most sensitive to changes in demand patterns. Figure 7 shows the number of required secondary sort hours under different robustness levels.

Figure 7: The number of required secondary sort hours for various robustness levels

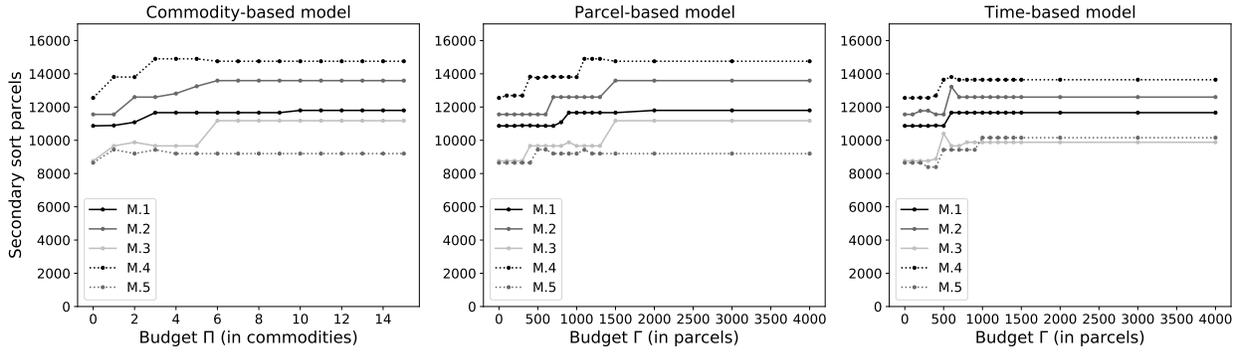


As expected, in all of the models, the number of secondary sort operational hours is non-decreasing as the value of the uncertainty budget increases. Operating plans with uncertainty budget set to zero, which corresponds to the nominal case, result in the least number of required secondary sort operational hours, i.e. they are least expensive in terms of operational cost. On the other hand, for plans with uncertainty budget greater than zero, extending secondary sort operational hours is often required to hedge against the specified level of uncertainty.

After optimizing for the secondary sort hours, we optimize for a secondary objective, the number of parcels that require secondary sorting. Figure 8 shows the value of this secondary objective for various robustness levels.

Since we optimize for the two objectives in a hierarchical fashion, higher values of uncertainty budgets do not necessarily result in non-decreasing values of the secondary objective we optimize for. This is not unexpected, since we add a new restriction when solving the second problem that is dependent on the first objective function value, which is non-decreasing in the value of the uncertainty budget, thus providing more relaxed restrictions for the second problem for higher uncertainty budgets. We next focus on the results generated by a couple of the instances to explain this further and discuss the main results that a decision maker would want to consider when using

Figure 8: The number of parcels requiring secondary sort for various robustness levels



any of the three proposed robust models.

Figures 9 and 10 show the results for instances M.1 and M.2, respectively. The proposed models enable decision makers to examine the price of robustness when generating sort plans under various types and levels of uncertainty. It is clear that there are considerable differences in the resulting sort plans for uncertainty budgets between the two extreme cases, which are easier to solve; this justifies the need for the robust models proposed in this work.

Figure 9: The number of secondary sort hours and parcels for instance M.1

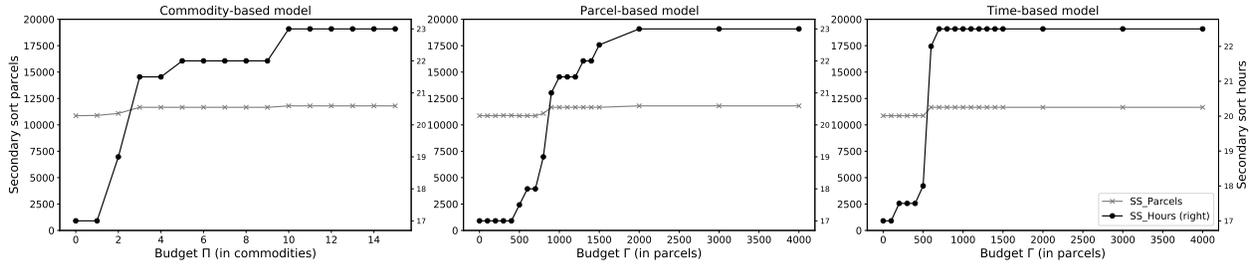
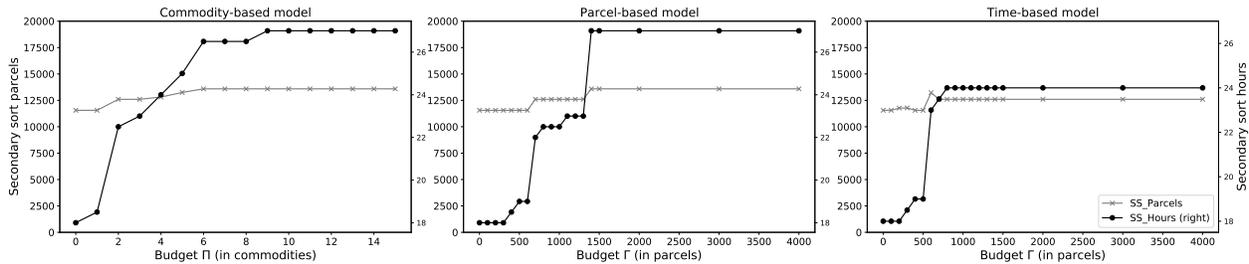


Figure 10: The number of secondary sort hours and parcels for instance M.2



By examining the results in Figure 9, a hub manager using the *parcel-based* model can evaluate plans generated at Γ values of 400, 500, 700, 800, 900, 1,200, 1,400, 1,500, or 10,000 parcels; the

remaining generated options have similar costs while providing less protection against uncertainty. To illustrate this further, Figure 11 shows the corresponding pile compositions for different Γ levels. Notice that even though pile assignments generated at $\Gamma = 400$ share the same cost with the nominal plan, the generated pile composition is different; plans generated at $\Gamma = 0$ are not guaranteed to be feasible for $\Gamma = 400$ while plans generated at $\Gamma = 400$ are guaranteed to be feasible for any $\Gamma \leq 400$. The same applies to plans generated at $\Gamma = 1,000$ and $\Gamma = 1,200$, where the latter is guaranteed to be feasible for all $\Gamma \leq 1,200$, while the reverse is not true.

Figure 11: (color online) Examples of pile assignments at different values of Γ , where each curve represents cumulative arrivals to a pile with specific deadline.

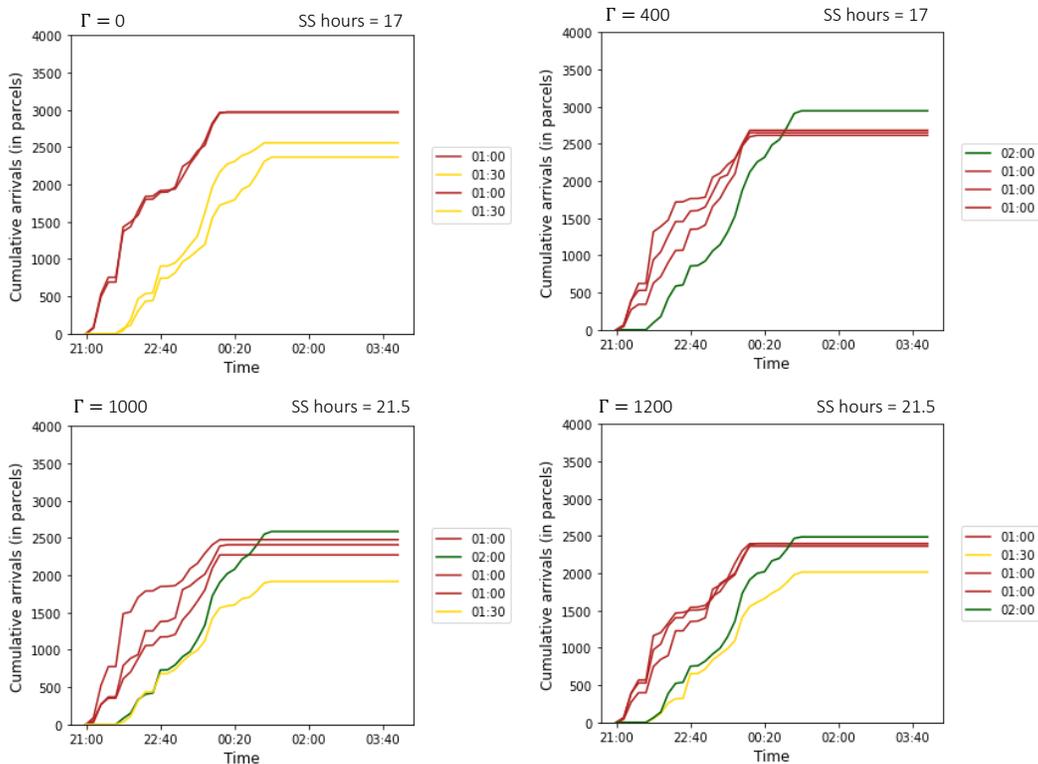


Table 3 provides a summary of the percent increase in operational requirements for various robust plans for instance $M.1$, when compared to its nominal plan requirements. As the table indicates, a hub manager can protect against up to 700 parcels changing from their nominal values (under the parcel-based uncertainty model) without the need to operate an additional secondary sort station or increase secondary sorters' workload; this is achieved by extending the operational hours of a secondary sorter that is already in use by one hour only. Furthermore, the plan at

Table 3: *The price of robustness for instance M.1 under the parcel-based model*

Budget	Secondary sort hours		Secondary sort parcels		Secondary sort stations	
	optimal value	% increase	optimal value	% increase	value	% increase
0	17	0.00	10867	0.00	4	0.00
400	17	0.00	10892	0.23	4	0.00
500	17.5	2.94	10867	0.00	4	0.00
700	18	5.88	10867	0.00	4	0.00
800	19	11.76	11086	2.02	4	0.00
900	21	23.53	11663	7.32	5	25.00
1200	21.5	26.47	11663	7.32	5	25.00
1400	22	29.41	11663	7.32	5	25.00
1500	22.5	32.35	11663	7.32	5	25.00
10000	23	35.29	11799	8.58	5	25.00

$\Gamma = 900$ requires the same number of secondary sort stations required to protect against the fully robust case. However, in the latter case, one of the secondary sorters needs to operate for two additional hours.

Similarly, Figure 10 summarizes the results for instance M.2. In this case, a hub manager using the time-based model is faced with five effective sort plan options, generated for Γ values of 200, 300, 500, 600, 700 or 10,000 parcels; the remaining options have similar costs while providing less protection against uncertainty. Table 4 provides a summary of the percent increase in operational requirements for various robust plans when compared to the nominal plan requirements.

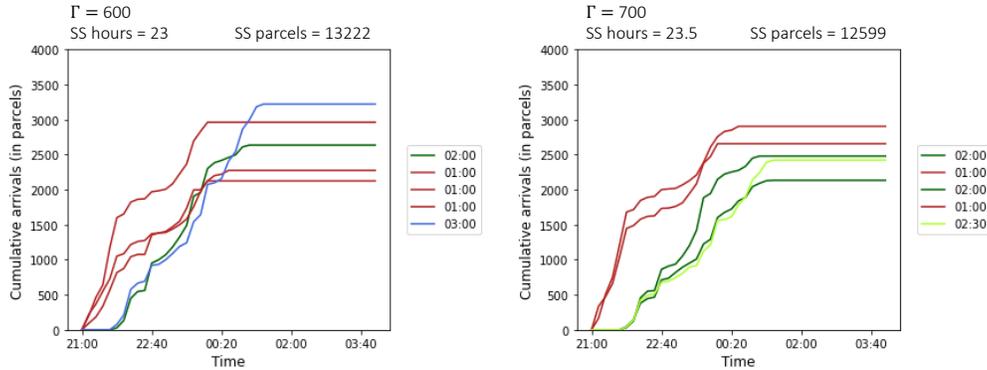
Table 4: *The price of robustness for instance M.2 under the time-based model*

Budget	Secondary sort hours		Secondary sort parcels		Secondary sort stations	
	optimal value	% increase	optimal value	% increase	optimal value	% increase
0	18	0.00	11558	0.00	4	0.00
200	18	0.00	11765	1.79	4	0.00
300	18.5	2.78	11785	1.96	4	0.00
500	19	5.56	11558	0.00	4	0.00
600	23	27.78	13222	14.40	5	25.00
700	23.5	30.56	12599	9.01	5	25.00
10000	24	33.33	12599	9.01	5	25.00

Notice that in this instance, operating plans that protect against higher levels of uncertainty could result in reduced secondary sort workload. For example, operating a plan that protects against up to 700 parcels arriving one hour later than their expected arrival times results in a 6% reduction in secondary sort workload when compared to plans that protect against up to 600

parcels arriving late; extending the operational hours by just 30 minutes changes pile compositions such that more parcels get assigned to direct sort. A detailed pile composition for $\Gamma = 600$ and $\Gamma = 700$ is shown in Figure 12.

Figure 12: (color online) Pile assignment by deadline for instance M.2 at $\Gamma = 600$ and $\Gamma = 700$. Notice that the number of parcels that require secondary sorting is higher at $\Gamma = 600$ than at $\Gamma = 700$ while the latter has slightly longer secondary sort hours.



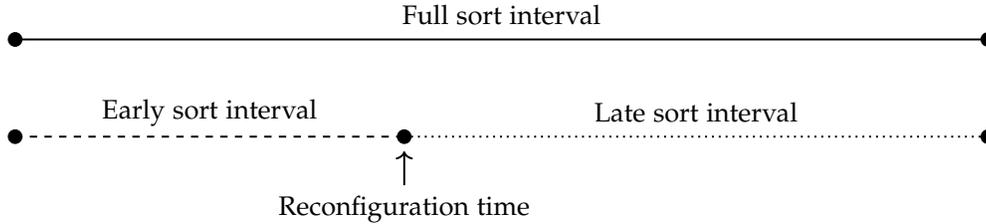
5.4 Other practical considerations

5.4.1 Allowing pile reconfiguration

So far, we have presented results for the original formulation, which generates assignments of commodities to primary sort piles that are fixed throughout a sort shift. This stable setting is generally preferred in practice to ease process control and gain operational efficiencies, especially when manual labor is involved. In this section, we explore the value of introducing more flexibility to sort operations by allowing pile assignments to change *once* during a sort shift. Specifically, we want to see if such limited flexibility can help reduce the price of robustness in the generated plans while protecting against various levels of uncertainty. To do so, we introduce a few modifications to the original formulation, as detailed in the Appendix. In this model variant, we allow a pile to be assigned a primary sort position p , a deadline c , a sort mode s and a sort interval i that defines either an early sort interval, a late sort interval or a full sort interval that corresponds to the full sort shift, as depicted in Figure 13.

Under this setting, a pile that is assigned to an early sort shift can be reused to process other commodities during the late sort shift. Parcels can be assigned to a sort interval as long as their commodity first arrival time and their deadline both fall within the sort interval time window.

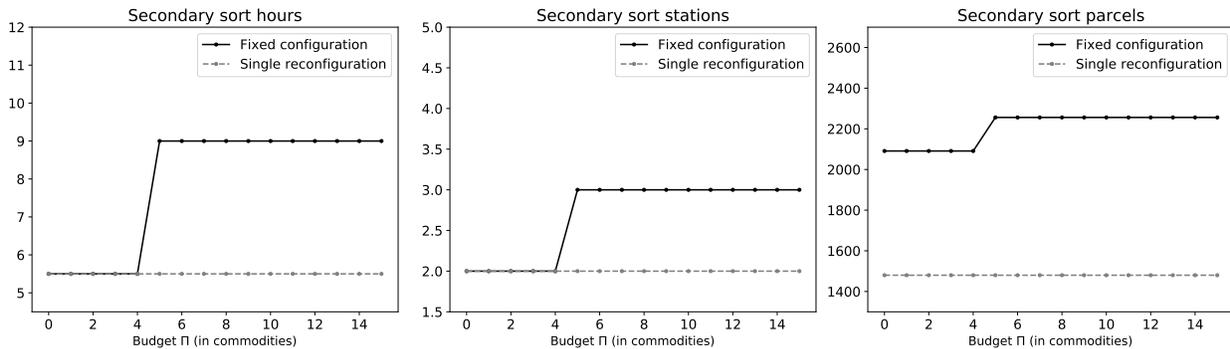
Figure 13: An illustration of our modeling extension idea that allows for pile to be reconfigured during a sort shift. A reconfiguration time in this setting is user-defined, i.e., it is an input to our problem.



Therefore, a reasonable choice of a reconfiguration time is one that coincides with one of the pile sort deadlines.

We test this variant for the commodity-based uncertainty model using the same test instances presented earlier. Our results show that Daysort instances benefit the most from pile reconfiguration, while Twilight and Midnight instances do not benefit as much. This can be explained by parcel arrival patterns, which tend to be more spread out over longer time spans during the Twilight and Midnight shifts, making assignments to early or late sort intervals less effective, and impossible in many cases. However, this variant is still useful for Daysort instances, characterized by smaller arrival quantities and a wider range of commodity types – typically parcels to be delivered to local and nearby destinations. Figure 14 shows the results for a Daysort instance. Recall a Daysort shift starts at 12PM and ends at 4PM; in this instance, the most effective reconfiguration time is found to be the one that coincides with the first sort deadline, which is 2PM.

Figure 14: Results of Instance D.1 using the commodity-based model variant, showing that we can generate fully robust plans at the cost of nominal plans when we allow pile assignments to be reconfigured once.



As the figures indicate, we can be fully robust (protect against the most conservative scenarios) at the nominal plan cost when we allow piles to be reconfigured once during a Daysort shift; a

single reconfiguration forestalls the need to extend operational hours, increase secondary sort workload or add extra secondary sorters. The very limited flexibility of allowing piles to be reconfigured once can help reduce the price of robustness while hedging against uncertainty; the presented variant allows generating more cost-effective alternatives to fixed sort plans that require higher operational costs to protect against various levels of uncertainty.

5.4.2 Considering other objective functions

The objective functions used in the case study were derived in consultation with our industry partner given their priorities. Other objectives could be used within this framework and might be more appropriate for other sort hub configurations and operating characteristics. For example, a planner might be interested in minimizing the number of required secondary sort stations; such an objective directly correlates with minimizing labor and machine costs when operating a hub with manual secondary sorting, and reducing fixed and operational costs associated with machines when operating a hub with fully automated secondary sorting. A planner may also be interested in optimizing for objectives that are not cost related. For example, balancing workload between sort stations is an important objective in facility logistics to improve resource utilization and reduce congestion.

To illustrate the flexibility of the proposed models to accommodate different objective functions, we extend our computational experiments to test the performance of these models when minimizing for the number of required secondary sort stations, modeled as

$$\text{Minimize } \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} y_{p,c,2}. \quad (15)$$

Figure 15 illustrates the resulting computational performance when using midnight shift instances while Figure 16 illustrates the resulting optimal objective function values as a function of budget levels.

Similar to the trends observed in Section 5.2, run times vary by model and budget level. It typically takes longer to solve for budgets that are not too small or too large, which is expected. All of the models, however, can be solved to find optimal solutions for various budget levels in a few seconds; the solver spends the remaining time proving solution optimality. Figure 16 results can be used by a planner to better understand the trade-offs between the desired robustness level

Figure 15: Run time to optimality for various robustness levels under the single objective of minimizing secondary sort stations

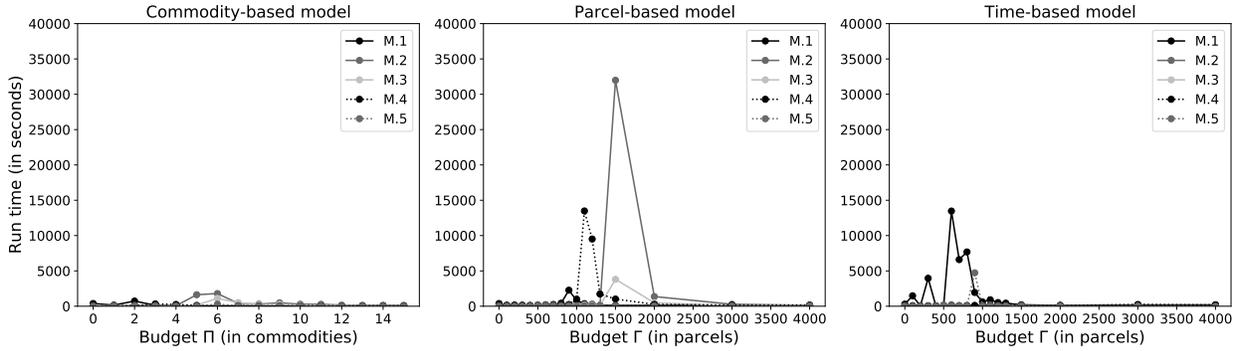
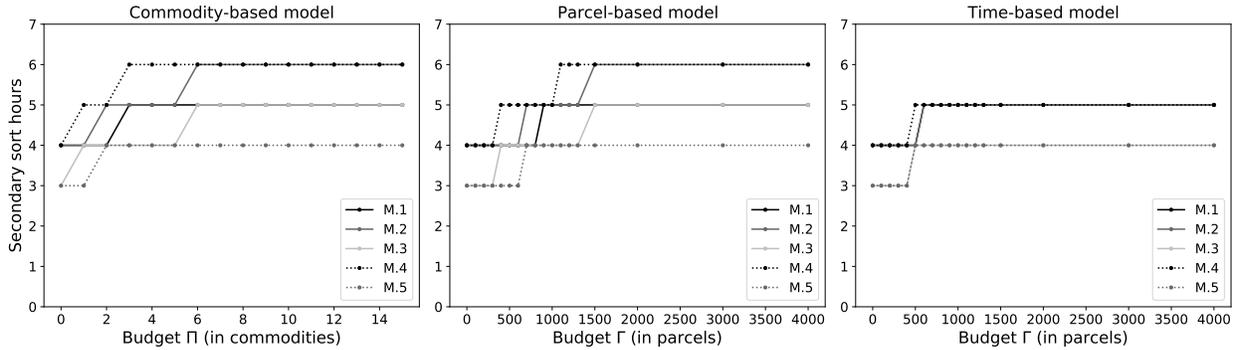


Figure 16: The number of required secondary sort stations for various robustness levels



and the corresponding number of secondary sorters required to complete parcel sorting in time. For instance, a planner operating instance M.4 and using the commodity-based model is able to identify a plan that protects against any single commodity having its arrival profile increased by at most 30% from expectation while requiring the same number of sorters as the nominal plan, i.e., at no additional cost, while two additional sorters are needed to operate a fully robust plan that protects against any commodity having its arrival profile inflated by at most 30% from its expected arrival profile in the worst case.

6 Conclusion

In this paper, we develop robust optimization models to generate sort plans that assign parcels to primary sort equipment operating within express parcel delivery sort hubs. Since frequently changing sort plans to address fluctuation in demand is not practical, it is important to generate assignments that take into account demand changes and uncertainty while being cost-effective and

feasible for longer periods of time. We propose models that explicitly incorporate various sources and levels of demand uncertainty commonly experienced in practice. Using practical insights and industry data, we propose different uncertainty models that take into account changes in arrival quantities and/or arrival times. We exploit problem structure to generate computationally tractable robust models that solve realistic-sized instances, with high-quality solutions obtained in relatively short CPU time. The proposed models can be used by hub managers to investigate trade-offs between plans' robustness and their associated operational costs, to ultimately operate sort plans that are protected against desired levels of robustness for improved system and service performances. The computational tractability of the proposed models allow us to extend our model to test other practical considerations. In particular, we modify our formulation for the commodity-based model to test the value of allowing piles to be reconfigured once during a sort shift. Our results show that such limited flexibility can help reduce the price of robustness, especially in Daysort shifts, which typically process small parcel volumes heading to a wide range of nearby and local destinations.

Future research avenues include integrating sort decisions with other upstream and downstream system decisions, including network-wide load planning decisions. Examples include building tractable models that optimize for the detailed sort process as well as inbound/outbound truck scheduling decisions. Another interesting research avenue is to explore the value of introducing more flexibility in the operational planning of sort hubs while investigating the impact they have on operational- and service-related performance measures. This includes: (1) moving from a static to a dynamic sort planning framework that makes use of real-time data related to actual parcel types, quantities, and arrival times and (2) introducing a more flexible worker schedule to improve worker utilization in cases where the secondary sort stage is manually operated.

References

- Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European journal of operational research*, 190(1), 1–21.
- Bakker, H., Dunke, F., & Nickel, S. (2020). A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice. *Omega*, 96, 102080.

- Baldi, M. M., Manerba, D., Perboli, G., & Tadei, R. (2019). A generalized bin packing problem for parcel delivery in last-mile logistics. *European Journal of Operational Research*, 274(3), 990–999.
- Bartholdi, J. J., & Gue, K. R. (2004). The best shape for a crossdock. *Transportation Science*, 38(2), 235–244.
- Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3), 464–501.
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations research*, 52(1), 35–53.
- Boysen, N., Briskorn, D., Fedtke, S., & Schmickerath, M. (2019). Automated sortation conveyors: A survey from an operational research perspective. *European Journal of Operational Research*, 276(3), 796–815.
- Boysen, N., Fedtke, S., & Weidinger, F. (2017). Truck scheduling in the postal service industry. *Transportation Science*, 51(2), 723–736.
- Boysen, N., Stephan, K., & Weidinger, F. (2019). Manual order consolidation with put walls: The batched order bin sequencing problem. *EURO Journal on Transportation and Logistics*, 8(2), 169–193.
- Briskorn, D., Emde, S., & Boysen, N. (2017). Scheduling shipments in closed-loop sortation conveyors. *Journal of Scheduling*, 20(1), 25–42.
- Çeven, E., & Gue, K. R. (2017). Optimal wave release times for order fulfillment systems with deadlines. *Transportation Science*, 51(1), 52–66.
- Chen, J. C., Chen, T.-L., Ou, T.-C., & Lee, Y.-H. (2019). Adaptive genetic algorithm for parcel hub scheduling problem with shortcuts in closed-loop sortation system. *Computers & Industrial Engineering*, 138, 106114.
- Crainic, T. G. (2000). Service network design in freight transportation. *European journal of operational research*, 122(2), 272–288.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4), 1096–1109.
- Fedtke, S., & Boysen, N. (2017). Layout planning of sortation conveyors in parcel distribution centers. *Transportation Science*, 51(1), 3–18.
- Gabrel, V., Murat, C., & Thiele, A. (2014). Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3), 471–483.

- Gallien, J., & Weber, T. (2010). To wave or not to wave? order release policies for warehouses with an automated sorter. *Manufacturing & Service Operations Management*, 12(4), 642–662.
- Haneyah, S., Schutten, J. M., & Fikse, K. (2014). Throughput maximization of parcel sorter systems by scheduling inbound containers. *Efficiency and innovation in logistics* (pp. 147–159). Springer.
- Haneyah, S., Schutten, J. M., Schuur, P., & Zijm, W. H. (2013). Generic planning and control of automated material handling systems: Practical requirements versus existing theory. *Computers in industry*, 64(3), 177–190.
- Jarrah, A. I., Qi, X., & Bard, J. F. (2016). The destination-loader-door assignment problem for automated package sorting centers. *Transportation Science*, 50(4), 1314–1336.
- Keith, A. J., & Ahner, D. K. (2019). A survey of decision making and optimization under uncertainty. *Annals of Operations Research*, 1–35.
- Khair, R., Erera, A., & Toriello, A. (2021). Two-stage sort planning for express parcel delivery. *IIEE Transactions*, 1–35.
- McWilliams, D. L. (2009). Genetic-based scheduling to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, 56(4), 1607–1616.
- McWilliams, D. L., & McBride, M. E. (2012). A beam search heuristics to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, 62(4), 1080–1092.
- McWilliams, D. L., Stanfield, P. M., & Geiger, C. D. (2005). The parcel hub scheduling problem: A simulation-based solution approach. *Computers & Industrial Engineering*, 49(3), 393–412.
- Novoa, L. J., Jarrah, A. I., & Morton, D. P. (2018). Flow balancing with uncertain demand for automated package sorting centers. *Transportation Science*, 52(1), 210–227.
- Research, & Markets. (2020). *Global parcel delivery market insight report 2020*. Retrieved October 27, 2020, from [https://www.researchandmarkets.com/reports/5018085/global-parcel-delivery-market-insight-report-2020?utm_source=dynamic&utm_medium=GNOM&utm_code=fsrl3l&utm_campaign=1381599++Global+Parcel+Delivery+Market+Insight+Report+\(2020+to+2024\)++Impact+of+the+COVID-19+Pandemic&utm_exec=jamu273gnomd](https://www.researchandmarkets.com/reports/5018085/global-parcel-delivery-market-insight-report-2020?utm_source=dynamic&utm_medium=GNOM&utm_code=fsrl3l&utm_campaign=1381599++Global+Parcel+Delivery+Market+Insight+Report+(2020+to+2024)++Impact+of+the+COVID-19+Pandemic&utm_exec=jamu273gnomd)
- Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6-7), 971–983.

- Werners, B., & Wülfing, T. (2010). Robust optimization of internal transports at a parcel sorting center operated by deutsche post world net. *European Journal of Operational Research*, 201(2), 419–426.
- Yu, X., Liao, X., Li, W., Liu, X., & Tao, Z. (2019). Logistics automation control based on machine learning algorithm. *Cluster Computing*, 22(6), 14003–14011.
- Zhou, C., Stephen, A., Cao, X., & Wang, S. (2021). A data-driven business intelligence system for large-scale semi-automated logistics facilities. *International Journal of Production Research*, 59(8), 2250–2268.
- Zi, L., & Gao, B. (2020). Layout planning problem in an innovative parcel sorting system based on automated guided vehicles. *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, 630–637.

7 Appendix

7.1 Parcel-time uncertainty model

In this model of uncertainty, every commodity $k \in \mathcal{K}$ can have its arrival profile scaled up or down by a fraction $\alpha \in [0, 1]$ in the worst case. Additionally, parcels that belong to commodity type k can be delayed to arrive after their expected arrival time by at most δ time units measured in the discretization of \mathcal{T} , in the worst case, as long as they arrive by their latest feasible arrival time ℓ^k . Changes in arrival quantities due to demand fluctuations or delays are limited to Γ parcels in total across all commodities, and total arrivals are not allowed to exceed the total nominal arrival quantities by more than a fraction $\Delta \in [0, 1]$ throughout a sort shift. Similar to our derivation of the models presented in the paper, one can derive the corresponding robust counterpart following a similar procedure, using the following linear adversary problem:

$$\beta_{t,p,c,2}(\Gamma, \bar{x}) = \underset{\lambda, \pi, \rho}{\text{Maximize}} \quad \sum_{k \in \mathcal{K}} \sum_{t < \tau \leq c} \alpha q_{\tau}^k \bar{x}_{p,c,2}^k (\lambda^k - \pi^k) + \sum_{k \in \mathcal{K}} \bar{x}_{p,c,2}^k \rho^k \quad (16a)$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (\lambda^k + \pi^k) + \sum_{k \in \mathcal{K}} \rho^k \leq \Gamma, \quad (16b)$$

$$\sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} \alpha q_{\tau}^k (\lambda^k - \pi^k) + \sum_{k \in \mathcal{K}} \rho^k \leq \Delta \left(\sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}} q_{\tau}^k \right), \quad (16c)$$

$$0 \leq \lambda^k \leq 1 \quad \forall k \in \mathcal{K} \quad (16d)$$

$$0 \leq \pi^k \leq 1 \quad \forall k \in \mathcal{K} \quad (16e)$$

$$0 \leq \rho^k \leq \sum_{t-\delta+1 \leq \tau \leq t} q_{\tau}^k \quad \forall k \in \mathcal{K} : \ell^k > t. \quad (16f)$$

7.2 Model extension to allow for pile reconfiguration

In this variation, we allow a pile to be reconfigured once during a sort shift. For this, we introduce \mathcal{I} to denote the set of sort intervals, where $i = 1$ indicates an early sort shift, $i = 2$ indicates a late sort shift, and $i = 0$ indicates a full sort shift. We modify all the decision variables by adding this additional index. To generate a commodity-based model that allow piles to be reconfigured, we need to replace constraints (2c) in (4) with:

$$\sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} y_{p,c,s}^i \leq |\mathcal{P}| - \sum_{p \in \mathcal{P}} \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} y_{p,c,s}^0 \quad \forall i \in \mathcal{I} \setminus \{0\}.$$

This is needed to ensure that at most $|\mathcal{P}|$ pile positions are used at any time during the sort shift. We also need to add the following constraints to ensure that a pile position is used during an early or late sort interval only if it is not used for a full sort interval.

$$y_{p,c,s}^i \leq (1 - y_{p,c,s}^0) \quad \forall i \in \mathcal{I} \setminus \{0\}, p, c, s.$$

A similar reformulation can be applied to the parcel- and time-based uncertainty models.