

This is the peer-reviewed, accepted version of the paper published by OR Spectrum (2025), 47 (3), 743–780. 10.1007/s00291-025-00810-5.

© 2025. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Last-mile logistics via cargo tunnel: Exact optimization approaches for the cargo tunnel scheduling problem

Celso Gustavo Stall Sikora¹, Nils Boysen^{2,*}, Dirk Briskorn³

October 2023

Revised: July 2024, November 2024, and January 2025

¹: Universität Hamburg

Institute for Operations Research

Moorweidenstr. 18, 20148 Hamburg, Germany

<https://www.bwl.uni-hamburg.de/or.html>

celso.sikora@uni-hamburg.de

²: Friedrich-Schiller-Universität Jena

Lehrstuhl für Operations Management

Carl-Zeiß-Straße 3, 07743 Jena, Germany

<http://www.om.uni-jena.de>

nils.boysen@uni-jena.de

*: Corresponding author, phone +49 3641 9-43100

³: Bergische Universität Wuppertal

Professur für BWL, insbesondere Produktion und Logistik

Gaußstr. 20, 42119 Wuppertal, Germany

<http://www.prodlog.uni-wuppertal.de>

briskorn@uni-wuppertal.de

Abstract

To relieve congested city centers, cargo tunnels are a recently revitalized last-mile delivery concept that is currently evaluated in various capability studies in different parts of the world. Under the cargo tunnel concept, freight is lifted into a tunnel at an easy-to-access outskirt depot and loaded onto electrically-powered autonomous cargo vehicles, rail cars, or maglev shuttles. These small shuttles move the goods underground through the tunnel toward small inner-city hubs, where environmentally-friendly vehicles such as cargo bikes take over the final transport leg toward urban customers. A central decision task in this context is the cargo tunnel scheduling problem, where a given set of shipments to be transported through the tunnel must be assigned to the departure slots of tunnel vehicles, such that neither vehicle capacities nor storage capacities in the hub are violated, all shipments reach their destination hubs in time, and the utilization of tunnel vehicles is minimized. This paper is dedicated to deriving exact solutions for this optimization problem. We present three alternative solution approaches, namely, two mixed integer programs and a Dantzig-Wolfe decomposition based on column generation and branch-and-price. An extensive computational study evaluates the performance of these exact solution methods. Furthermore, we show that a lack of shipment consolidation, long unloading times, and scarce storage capacities within the inner-city hubs bear the risk that significant parts of the tunnel capacity remain unused.

Keywords: Smart city logistics; Last-mile delivery; Cargo tunnel; Scheduling.

1 Introduction

During their endeavors of becoming “smart cities”, many urban communities seek solutions for reducing freight traffic and the negative effects on health, congestion, and greenhouse gas emissions coming along with it. One innovative solution in this regard, currently revitalized in quite a few capability studies all around the globe, are cargo tunnels. Tunnel-based transportation systems have a long-lasting tradition in public (passenger) transport. The eldest urban underground railway system, London subway, exists since 1863 [43], and nowadays subway systems are inevitable transportation backbones of most major cities. Many subway systems already operate at their capacity limits, and existing train as well as subway station infrastructure is often ill-suited for moving (a considerable amount of) freight in parallel to their human passengers [13]. Hence, extra cargo tunnels specifically dedicated to freight are considered as an alternative.

Cargo tunnels, too, have a surprisingly long tradition. It started in 1853 with a pneumatic dispatch system for moving telegrams between the offices of Electric and International Telegraph Co. with the London Stock Exchange [41]. In the following decades, these systems became very popular and were erected all over the world [37]. Larger pneumatic systems have also been applied for wheeled capsules, mainly for moving gravel or waste, in the UK, Russia, and Japan [41]. Based on a different technology, rail-bound systems have then evolved at the beginning of the 20th century.

Among the most prominent are the underground rail transport system of Chicago [27] and the London Post Office Railway [4]. Both systems operated until 1959 and 2003, respectively.

Analogously to the reinvention of high-speed passenger travel within tubes, for which first ideas date back to the 19th century [3] and new plans got promoted by Elon Musk's Hyperloop project [30], also cargo tunnels received renewed attention in the recent years. In the following, we briefly elaborate a few capability studies for tunnel-based freight logistics that have been initiated in the recent years in different parts of the world [for more projects, see 41]:



Figure 1: HyperPort concept of HyperloopTT and HHLA (left; Source: HHLA/Mormedi) and Swiss cargo tunnel project Cargo Sous Terrain (right; Source: CST)

- A project called “Smart City Loop” promotes a tunnel project below river Elbe into the city center of Hamburg. The tunnel is projected with a length of 5 km tunnel, a diameter of 4 m, and a conveyor system that moves palatalized goods from an outskirts depot to a single inner-city micro hub. After a successful feasibility study, the planning phase for the realization of this project was initiated at the end of 2020 [24].
- An even more ambitious project is “Cargo Sous Terrain” (see Figure 1(right)). Under this name, Switzerland plans a tunnel system of 490 km that connects most major Swiss cities until 2031. The tunnel is projected with a diameter of 6 meters and three travel lanes, on which autonomous electrified cargo shuttles operate on a paved surface at 30 km/h [2].
- A British project, promoted under Magway.com, proposes a maglev monorail system within under-ground and above-ground pipelines with a diameter of 0.9 meters. Small shuttles, propelled by linear synchronous magnetic motors, are planned to operate at 50 km/h.
- In cooperation with German port operator HHLA, HyperloopTT promotes a concept called “HyperPort” for the port of Hamburg (see Figure 1(left)). Here, containers are planned to be moved into the hinterland in large freight capsules at 600 km/h [22]. Other feasibility studies for a tunnel-based container movement to and from ports were, for instance, presented for Shanghai [19] and Antwerp [40].

The main justification for these projects and the huge tunnel boring costs coming along with their realization, is the reduction of surface traffic and all the negative impact on health, safety, congestion, and the environment coming along with it. When calculating this advantage, however,

most feasibility studies presuppose a tunnel operating at full capacity at all times. For the Smart City Loop in Hamburg, for instance, this leads to a total tunnel capacity of 2.7 million pallets per year, which is equivalent to 540.000 fewer truck drives into the city center and 10,000 tons of saved carbon dioxide emissions [24].

In a recent paper, however, Boysen et al. [8] have shown that time-critical shipments in combination with capacity-restricted inner-city hubs are obstacles when it comes to actually loading the tunnel and its shuttles to capacity. They identify situations where large parts of the tunnel capacity remain unused. Their finding, however, is based on a simplified version of the cargo tunnel scheduling problem (CTSP) that presupposes rather strong assumptions, i.e., all shipments have the same size and unloading shuttles in micro hubs takes no time. This allows them to derive a basic CTSP version that is solvable to optimality in polynomial time. Our paper is also dedicated to the CTSP, which can be characterized as follows: Given a set of shipments to be transported through the tunnel, this problem must assign shipments to departure slots of tunnel vehicles, such that neither transport capacities of shuttles nor storage capacities in hubs are violated, the shipments' due dates at their destination hubs are met, and the utilization of tunnel vehicles is minimized. We also aim at optimal solutions for this problem; other than Boysen et al. [8], however, we target a more realistic problem setup with differently-sized shipments, non-zero unloading times in hubs, and a more detailed objective function. Based on a detailed analysis of computational complexity, we show that these features make the cargo tunnel scheduling problem a challenging combinatorial optimization problem.

To solve this problem to optimality, we introduce three exact solution approaches, namely two mixed integer programs (MIP) and a Dantzig-Wolfe decomposition procedure based on column generation and the branch-and-price paradigm. Unfortunately, our computational study shows that the Dantzig-Wolfe decomposition is outperformed by its competitors. Hence, we present the details of this method only in the appendix. However, to warn other researchers of our laborious, yet unsuccessful efforts, we keep the computational results of this approach in the paper. After benchmarking our solution methods according to their computational performance, we apply the best solution method to challenge the findings of Boysen et al. [8] within a more realistic problem setting. We, too, identify situations where large parts of the tunnel capacity must remain empty to not violate capacity and/or delivery time restrictions. Other than Boysen et al. [8], however, we show that the problem of wasted tunnel capacity can be even more severe when additional constraints of the knapsack type restrict the loading of tunnel vehicles. Thus, this paper adds the following contributions to the cargo tunnel literature:

- We formulate a more realistic cargo tunnel scheduling problem that steers the movement of shipments through the tunnel under less restrictive assumptions than previous research.
- We provide a thorough analysis of the computational complexity.
- We present three novel exact solution approaches for solving the cargo tunnel scheduling problem to optimality.

- Finally, we apply our best solution method to investigate managerial issues related to cargo tunnels. We identify situations where significant parts of the tunnel capacity remain unused. Furthermore, we show that consolidation of shipments, fast handling times, and enough storage capacity within the inner-city hubs are impactful levers to avoid this problem.

The remainder of the paper is structured as follows. Section 2 reviews the related literature. Section 3 is dedicated to the definition of our version of the CTSP. This section also contains our analysis of computational complexity. Section 4 and the appendix present our three exact solution approaches. Specifically, the section provides two MIPs (see Sections 4.1 and 4.2), while the Dantzig-Wolfe decomposition approach can be found in the appendix. Our computational study of Section 5 benchmarks the computational performance of our three solution approaches and addresses the managerial issues discussed above. Finally, Section 6 concludes the paper.

2 Literature review

Tunnel-based urban freight logistics belong to the field of (innovative) last-mile delivery concepts. How to supply the inhabitants of future smart cities is a vivid field of research, for which we, however, only refer to the latest literature surveys of Savelsbergh and Van Woensel [35], Speranza [36], and Boysen et al. [6]. Instead, we focus our literature review on (i) cargo tunnels in general and (ii) operations research procedures dedicated to freight tunnels in particular. Furthermore, we address (iii) research on delivery concepts that include public transport (mainly subways or trams) and (iv) research on related scheduling problems with a similar structure from other domains.

(i) Survey papers on underground traffic in general (i.e., also including peoples transport) and on the history as well as the status quo of freight tunnel projects are provided by Cui and Nelson [13] and Visser [41], respectively. Furthermore, there are some case studies that discuss the advantages and disadvantages of specific tunnel projects. Examples are provided by Hai et al. [19] (container tunnel for the port of Shanghai), Hu et al. [21] (integration of freight transport into an existing metro network), and Vernimmen et al. [40] (different container tunnel setups for the port of Antwerp). All three studies calculate with a fully utilized tunnel capacity.

(ii) There are only a few papers on solution methods for operating a cargo tunnel. A related research stream focuses on network design and the question where to place either a single tunnel or a network of tubes [e.g. 20, 29, 31, 42]. We, however, presuppose a given tunnel layout that leads from an outskirt depot along a given set of inner-city hubs at given positions. Another stream of the literature addresses operational decision problems but for pipeline-based transportation of non-discrete goods [e.g. 23, 26, 44]. To the best of the authors' knowledge, there are only two other papers directly in our area. Boysen et al. [7] assign unit-load shipments to departure slots of shuttles and coordinate the stream of goods through the tunnel with the delivery tours of a cargo bike at a single hub. Thus, they propose a two-echelon setup where other than in this paper the delivery person is explicitly modeled. Their study shows that small hub capacities increase the

travel distances of the delivery person considerably. Similar to the paper on hand, Boysen et al. [8] also treat the CTSP, where time-critical, less-than-shuttle-load shipments are assigned to shuttle departures toward multiple capacity-restricted hubs. To derive a polynomially-solvable problem version of the CTSP, however, they introduce some strong assumptions, such as all shipments having identical size, the exclusion of (un-)loading times, and a simplified objective function. This paper, instead, relaxes these strong assumptions, which leads to a much more involved, $\mathcal{NP}$ -hard problem version of the CTSP. In a real-world urban context, shipments of various sizes should be the norm, and neglecting unloading times can lead to infeasible schedules. Thus, our extended CTSP considers important real-life features. Moreover, Boysen et al. [8] only consider the shuttles' utilization when departing from the depot. However, if many shipments are unloaded at the first hubs, even a shuttle departing fully loaded can pass large parts of the tunnel almost empty. Thus, Boysen et al. [8] underestimate the threat of unused tunnel capacity, whereas our new objective function, based on the exact vehicle load at each hub, allows for a more realistic evaluation.

(iii) To save the huge tunnel boring costs, freight transport can also be combined with existing public transport lines, such as subways or trams. An overview on pilot studies in different cities of the world is provided by Hu et al. [21]. The recent survey paper of Cavallaro and Nocera [11], however, reveals that most studies in this area are rather of a qualitative nature. An exception is, for instance, the paper of Bruzzone et al. [9] who provide some simple analytical formulae to estimate the impact of an integrated passenger and freight transport. Other researchers extend vehicle routing problems with the option to hand some shipments over to crowdshippers who apply public transport on scheduled lines [e.g. 15, 16, 17]. The resulting problems settings, however, are completely different, so (to the best of the authors' knowledge and supported by the recent survey paper of Le et al. [25]) no existing approach of this area can directly be applied to our setting.

(iv) Assigning differently-sized shipments to capacity restricted shuttles resembles the famous bin packing problem [for a survey see 12]. Due dates and multiple subsequent micro hubs to be visited, however, extend this core problem considerably. These extension are rather considered within other logistics and vehicle scheduling problems on the line [e.g. 5, 18, 32, 38]. These studies, however, only consider the scheduling of the vehicles and do not include the assignment of time-critical, less-than-shuttle-load shipments to vehicles.

It can be concluded that previous research has not considered our extended version of the CTSP yet. Thus, we are the first to present exact solution approaches for a realistic tunnel scheduling problem including differently-sized shipments and non-zero unloading times.

3 Problem description

This section is dedicated to the problem version of CTSP treated in this paper. Here, we take the perspective of a single logistics company that aims to transport a given set of shipments through

the tunnel. To do so, the *shipper* (i.e., the logistics company) contacts the tunnel *operator* who is in charge of the tunnel infrastructure and rents out departure slots of tunnel *shuttles*. The assignment of shipments to available shuttle slots is the main decision task of the CTSP. We start with an intuitive elaboration of this problem, give an example, and discuss our simplifying assumptions in Section 3.1. A precise problem definition of CTSP and a thorough analysis of its computational complexity are provided in Section 3.2 and 3.3, respectively.

3.1 Problem characterization, example, and assumptions

The main entity of our CTSP is a line-shaped cargo tunnel, which connects an outskirts depot where all shuttle transports originate with a set of inner-city micro hubs with given positions along the tunnel. The tunnel infrastructure is maintained by an operator, who organizes the shuttle transports at given departure time slots. Furthermore, we have shippers, who aim to utilize the tunnel for servicing their own urban customers from the micro hubs. We assume that one shipper at a time contacts the operator and is offered the departure slots for shuttles that are still available. The latter means that these slots have not been hired by other shippers during previous planning runs yet. Now, it is the decision of the current shipper which of the still available slots they should hire. To do so, we suggest using CTSP solutions. Naturally, this decision is guided by the aim of the shipper to minimize the costs for the tunnel services of the operator, while all goods of the shipper can be feasibly transported.

Thus, the shipper's *decision task* is a selection among the available shuttle departure slots and an assignment of the given set of less-than-shuttle-load shipments, dedicated to the shipper's own urban customers, to the selected shuttle slots. These decisions face the following *constraints*. Each customer shipment has a size and a due date. The latter specifies when the shipment is picked up at its designated destination hub to be finally delivered to its customer. (i) Thus, it is to be ensured that the subset of shipments assigned to the same departure slot do not violate the shuttle's loading capacity. We assume that (once started) shuttles move with constant velocity through the tunnel and that unloading at a hub, if some shipment of the shuttle load is dedicated to a hub at all, takes constant time. Note that these and further assumptions are discussed below. Thus, knowing the departure time of the shuttle a shipment is assigned to and the other goods that are also loaded onto the same shuttle (and thus potentially require intermediate stops at previous hubs for unloading) allows us to calculate the arrival time of each shipment at its designated hub. (ii) These arrival times may not violate each shipment's given due date in order to not delay the final delivery from the hub to the customers. Finally, all shipments that have already arrived at their designated hub but have not been picked up yet (because their due dates are not yet reached) need to be stored at the hub. (iii) Since land within an urban area is costly, we assume limited storage space at each hub, which must thus not be violated by the total size of intermediately stored shipments at any time.

Given these constraints, our decision task is also guided by an *objective*. We assume that the

price demanded by the operator for the transport services will depend (a) on the number of slots that the shipper wants to hire and (b) on the total time each hired shuttle is occupied, which is determined by the travel time to the farthest destination hub plus unloading times. In this scheme, the operator thus prices the duration of shuttle utilization. However, to make the optimization task, which is shown to be a great challenge especially for an exact solution approach (see Section 3.3), a bit more handy, we assume that with regard to (a) the maximum number of slots that are allowed to be utilized is already given and we only minimize (b), the total shuttle occupation time. Since each hired departure slot blocks tunnel capacity for subsequent logistics companies, it is to be expected that the main cost driver is the number of slots hired whereas the shuttle occupation time is subordinate. This allows us to put both cost factors (a) and (b) in a lexicographic order, so that a feasibility version of our problem setting can be applied in a binary search framework to, first, minimize the number of utilized slots. Once this minimum is known, we can fix the number of allowed departure slots to this minimum and minimize subordinate cost factor (b). Thus, the objective of our optimization problem only covers (b) and considers (a) by additional constraints (iv) that limit the maximum number of departure slots that are allowed to be utilized.

Once the decisions of the current shipper are taken and contracted by the operator, the requested shuttle departure slots and the hub storage capacities are blocked and no longer available for subsequent planning runs, where further shippers seek tunnel transport services.

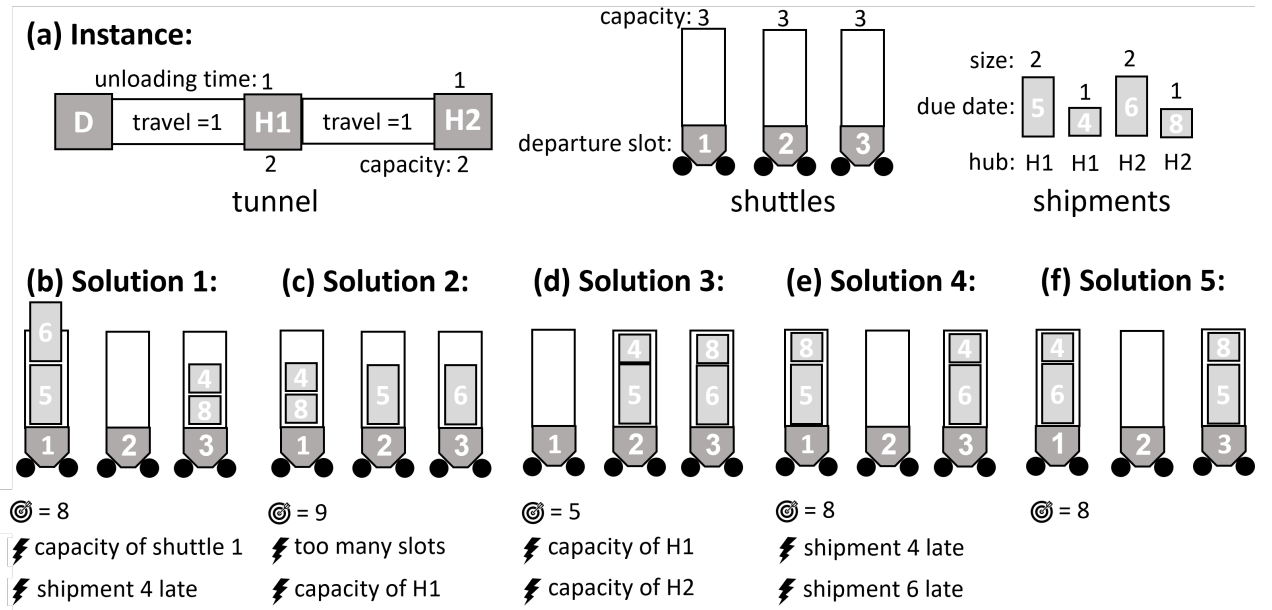


Figure 2: Example instance of our cargo tunneling problem with four solutions

Example: Consider the example instance depicted in Figure 2(a). Here, we have a cargo tunnel, which connects depot D with two inner-city hubs H1 and H2. The travel times between subsequent stations in the tunnel is 1 time unit, unloading takes 1 time unit in each hub, and the storage capacities in each hub are 2. The shipper is offered three departure slots from the depot at times 1, 2, and 3, where identical shuttles with a load capacity of 3 operate. The shipper has

to transport four shipments, whose destination hubs, due dates, and sizes are given in (a). Note that we identify shipments via their unique due dates, given by the white numbers within the gray shipment bars. Further note that the heights of the gray shipment bars indicate the sizes of the respective shipments. It takes at least two shuttles to feasibly transport these shipments, so that this number constitutes the example's maximum number of slots that are allowed to be occupied. Figures 2(b) to 2(f) depict five solutions, four of them (i.e., solutions 1 to 4) are infeasible and solution 5 is optimal. The former violate (i) the maximum load capacity of a shuttle (solution 1), (ii) lead to a late arrival of shipments (solutions 1 and 4), (iii) violate the limited hub capacity (solutions 2 and 3), or (iv) exceed the given maximum slot number (solution 2). We assume that no hub capacity is utilized during unloading, but once a shipment is stored in the hub it occupies storage space, also during the time slot of the due date. In solution 3, the shuttle of departure slot 3 is exclusively loaded with shipments for hub H2. Consequently, no stop (and no unloading time) at hub H1 is required, so that the shuttle already arrives at time 6 at hub H2. At time 6, both shipments with space requirement 3 thus exceed the storage capacity, although shipment 6 leaves the hub because its due date is reached. Only solution 5 of Figure 2(e) is feasible. Here, the two shipments loaded in slot 1 have already left their respective hubs H1 and H2, once the shuttle of slot 3 delivers further shipments to both hubs. Since both shuttles have to go all the way to H2 with intermediate unloading in H1, the total shuttle occupation time is 8, which is also the optimal objective value for this example.

Compared to the previous research of Boysen et al. [8], we aim at a more realistic version of the CTSP by considering differently-sized shipments and unloading times in hubs. However, our CTSP is still based on the some (simplifying) assumptions and prerequisites, which we list in the following:

- We only consider a unidirectional tunnel that moves freight from the depot into the city center. We assume that reverse movements out of the city are processed in another tunnel tube or lane, not considered in the CTSP. Reverse movements, e.g., waste or returned goods, are often less urgent. Based on this prerequisite, it seems reasonable to plan the city's supply via the tunnel first and use the selected shuttle drives (inducing the returns to the depot) as a given input for planning the reverse flow through the tunnel. We leave more involved settings where forward and backward flows influence each other to future research.
- We presuppose that a one-dimensional measure of capacities (i.e., of shipments, vehicles, and micro hubs) is sufficient to capture scarcity of space. We leave even more involved settings with more dimensions to future research.
- Shuttles move at constant speed through the tunnel. Slower travel or voluntary waiting (e.g., to relieve storage capacity at a hub) is assumed to impair other shuttles and is thus forbidden.

- If at least one shipment of a shuttle is dedicated to a hub, this hub must be visited and a constant unloading time is required. Voluntary hub visits and waiting at hubs is excluded. Note that variable unloading times that depend on the shipments that are unloaded (e.g., their number or sizes) are certainly also possible. However, we assume that most time is rather consumed by a shuttle to access a hub via switches, whereas the unloading process itself should be (largely) automatable (see the ideas for a fully automated (un-)loading process of shuttles presented at magway.com).
- Once a shipment has arrived at its dedicated hub and after it is unloaded there, it is assumed to occupy storage space. The latter holds even in time units where the due date is met.
- We have deterministic knowledge on all data, and all data can be expressed as integers.
- Finally, it is our objective to minimize the total shuttle occupation time, whereas the maximum number of shuttles that can be utilized is given as a constraint (see our elaboration above).

The next section is dedicated to a precise problem definition of CTSP.

3.2 Problem definition

The cargo tunnel is represented by line of length L with a depot at the start and a set H of inner-city hubs positioned along the line. Each position of the line is addressed by a number in $[0, L]$. The depot is located at position 0 and each hub $h \in H = \{1, \dots, |H|\}$ has a strictly positive integer position $0 < p_h \leq L$ and an individual storage capacity of C_h^H . We assume that hubs are numbered in increasing distance to the depot, that is $p_h < p_{h+1}$ for each $h = 1, \dots, |H| - 1$. Furthermore, we have a set $I \subseteq \mathbb{N}$ of shuttle departure slots offered to the shipper. The standardized tunnel shuttle of each departure slot has a load capacity of C^V . We refer to each departure slot $i \in I$ by the point of time i the associated shuttle leaves the depot. Note that since some of the actual departure slots have already been promised to other shippers during previous planning runs, the slot set I offered to the current shipper needs not refer to subsequent time slots. Each shuttle, once started from the depot, either is present at a hub for delivering a subset of loaded shipments, which takes constant unloading time P , or travels toward position L . Travel speed $1/T$ is constant, such that it takes each shuttle T time units to travel a distance of 1.

We, furthermore, consider a given set $S = \{1, \dots, |S|\}$ of shipments, where each shipment $s \in S$ has a destination hub $h_s \in H$ and a due date d_s , which defines the point of time when it is picked up at its destination hub and thus the latest point of time it can feasibly reach h_s . Furthermore, a shipment has a size c_s that reflects the amount of capacity a shipments occupies when carried by a shuttle or when stored in a hub. We consider a shipment to be finally delivered to its destination hub at the end of unloading time P after the shuttle reaching the respective hub.

A solution is an assignment a of each shipment to exactly one departure slot with $i_s^a \in I$ being the departure slot shipment s is assigned to. Let $I^a \subseteq I$ be the subset of slots having assigned at least one shipment, let S_i^a be the subset of shipments assigned to slot i according to a , and let, finally, H_i^a be the set of hubs the shuttle starting in slot i delivers at least one shipment to. A solution implies the point of time C_s^a shipment s reaches its destination hub h_s as

$$C_s^a = i_s^a + T \cdot p_{h_s} + P \cdot |\{h \mid h \in H_{i_s^a}^a, p_h \leq p_{h_s}\}|,$$

that is the start time i_s^a of its assigned departure slot plus total travel time $T \cdot p_{h_s}$ from the depot to h_s plus the total duration of intermediate stops before reaching h_s and unloading time at hub h_s . Furthermore, a solution implies for each shipment the set of shipments $\bar{S}_s^a$ that have destination hub h_s , are unloaded at h_s no later than d_s , and have due dates not earlier than d_s . Note that $s \in \bar{S}_s^a$. A solution a is feasible, if

- (i) the total size of loaded shipments of neither shuttle exceeds its available load capacity, that is $\sum_{s \in S_i^a} c_s \leq C^V$ for each $i \in I$,
- (ii) each shipment is unloaded at its destination hub no later than its due date, that is $C_s^a \leq d_s$,
- (iii) the number $|I^a|$ of hired shuttles does not exceed given threshold C^I , and
- (iv) the total size of stored shipments at no hub exceeds its available storage capacity at any point of time. Here, it suffices to focus on points of time where at least one shipment is picked up. Thus, we require $\sum_{s' \in \bar{S}_s^a} c_{s'} \leq C_{h_s}^H$ for each $s \in S$.

We evaluate each solution a by the total time O^a shuttles are occupied, that is $O^a = \sum_{i \in I^a} \max\{C_s^a - i_s^a \mid s \in S_i^a\}$. The CTSP is to find a feasible solution a , such that O^a is a minimum among all feasible solutions.

3.3 Analysis of computational complexity

Our first two results state that it is inherently hard to determine feasible solutions to CTSP, if $P > 0$ or we have to transport differently-sized shipments.

Theorem 1. *Determining whether a feasible solution to CTSP exists is strongly $\mathcal{NP}$ -hard, even if $I = \{0, \dots, |I| - 1\}$, $P = 0$, $C^I = C^H = \infty$, and $d_s = \infty$ for each $s \in S$.*

Theorem 2. *Determining whether a feasible solution to CTSP exists is strongly $\mathcal{NP}$ -hard, even if $c_s = 1$ for each $s \in S$, $C^I = \infty$, and $I = \{0, \dots, |I| - 1\}$.*

We abstain from formal proofs since they would be strictly analogue to those in Boysen et al. [8]. As opposed to the problem setting treated by Boysen et al. [8], however, CTSP remains hard to solve even if $P = 0$ and we only transport unit-load shipments of sizes $c_s = 1$.

Theorem 3. *CTSP is strongly $\mathcal{NP}$ -hard, even if $I = \{0, \dots, |I| - 1\}$, $P = 0$, $C^V = 3$, $C_h^H = 1$ for each $h \in H$, $C^I = \infty$, and $c_s = 1$ for each $s \in S$.*

The proof is by reduction from MAXBATCHING, which has been proven to be strongly $\mathcal{NP}$ -complete by Nonner [28].

MAXBATCHING: Given an interval graph $G = (V, E)$ with a weight w_v for each interval $v \in V$ and a number K find a partition of V into cliques $C_1, C_2, \dots, C_Q$ of at most size K such that total maximum weight $\sum_{q=1}^Q \max\{w_v \mid v \in C_q\}$ in cliques is minimized.

While the problem statement of Nonner [28] allows w_v , $v \in V$, to be rational, the proof of complexity shows that the problem is strongly $\mathcal{NP}$ -hard for integer weights and $K = 3$. Hence, we assume that $w_v \in \mathbb{N}$ for each interval $v \in V$.

Proof. Given an instance $\mathcal{I}$ of MAXBATCHING by interval graph $G = (V, E)$ with weight $w_v \in \mathbb{N}$ and associated intervals $[l_v, r_v[$ for each node $v \in V$, we, first, modify $\mathcal{I}$ in two steps. First, we multiply each weight by $|V|^2$ and make all weights different by increasing them, such that $k > 1$ nodes with former equal weight w now have weights $|V|^2 \cdot w, \dots, |V|^2 \cdot w + k - 1$. It is not hard to see that a solution has total maximum weight of W according to original weights, if and only if it has a total maximum weight of at least $|V|^2 \cdot W$ and less than $|V|^2 \cdot (W + 1)$ according to new weights. Second, we modify each interval by increasing l_v and r_v by $\lceil |V|/2 \rceil + \max\{w_v \mid v \in V\}$ and multiplying them with a large value afterwards. Clearly, this does not change whether or not two intervals overlap. However, no interval's left end point is smaller than $\lceil |V|/2 \rceil + \max\{w_v \mid v \in V\}$ and, if the value is chosen sufficiently large, each subset of overlapping intervals overlaps over a length of at least $|V|$ now. We refer to the modified instance as $\bar{\mathcal{I}}$ and conclude that there is a feasible solution to $\mathcal{I}$ with total maximum weight of exactly W , if and only if there is a feasible solution to $\bar{\mathcal{I}}$ with total maximum weight of at least $|V|^2 \cdot W$ and less than $|V|^2 \cdot (W + 1)$. We assume that in $\bar{\mathcal{I}}$ nodes in V are numbered according to increasing weights and construct the following instance $\mathcal{I}'$ of CTSP from $\bar{\mathcal{I}}$.

We consider $I = \{0, \dots, \max\{r_v \mid v \in V\}\}$ and $H = \{1, \dots, |V| + 1\}$ with position $p_h = w_h$ for $h = 1, \dots, |V|$ and $p_{|V|+1} = p_{|V|} + 1$. Hubs have a capacity of $C^H = 1$. Hub $|V| + 1$ receives $\lceil |V|/2 \rceil$ shipments with due dates $p_{|V|+1}, \dots, p_{|V|+1} + \lceil |V|/2 \rceil - 1$. Each other hub $h = 1, \dots, |V|$ receives two shipments s_h^1 and s_h^2 with due dates $d_{s_h^1} = l_h - 1 + p_h$ and $d_{s_h^2} = r_h - 1 + p_h$. Finally, we set $C^V = 3$ and $C^I = \infty$. The reduction is pseudo-polynomial.

We claim, then, that there is a feasible solution to $\mathcal{I}'$ with total occupation time of $\lceil |V|/2 \rceil \cdot (p_{|V|} + 1) + O$, if and only if there is a solution to $\bar{\mathcal{I}}$ with total maximum weight of O .

Assume that there is a feasible solution to $\mathcal{I}'$ with total occupation time of $\lceil |V|/2 \rceil \cdot (p_{|V|} + 1) + O$. Due to the position and capacity of hub $|V| + 1$ and its shipments' due dates, each vehicle starting in slots $0, \dots, \lceil |V|/2 \rceil - 1$ carries a single shipment to hub $|V| + 1$. We refer to these vehicles as fixed vehicles in the following. Further, since $d_{s_h^1} < d_{s_h^2}$ we can assume that for each hub $h = 1, \dots, |V|$ shipment s_h^1 is assigned to a vehicle starting not later than the one s_h^2 is assigned

to due to a simple interchange argument. Then, no second shipment for any hub $h = 1, \dots, |V|$ can be carried by fixed vehicles since s_h^1 blocks the hub's capacity such that s_h^2 can reach h only after $d_{s_h^1}$ and, hence, cannot leave the depot before $d_{s_h^1} + 1 - p_h = l_h \geq \lceil |V|/2 \rceil$. Then, we can, additionally, assume that the first shipment s_h^1 for hub $h = 1, \dots, |V|$ is assigned to a fixed vehicle since h has sufficient capacity, fixed vehicles have sufficient total capacity, first shipments do not compete with second shipments for vehicle capacity, and no additional occupation time of any fixed vehicles is imposed (recall that fixed vehicles head for the farthest hub $|V| + 1$ anyway). Concluding, first shipments to hubs $1, \dots, |V|$ and shipments to hub $|V| + 1$ imply total occupation time of $\lceil |V|/2 \rceil \cdot (p_{|V|} + 1)$ of fixed vehicles. This leaves total occupation time of O of vehicles that are not fixed for delivering second shipments to hubs $1, \dots, |V|$. The feasible interval of vehicles shipment s_h^2 can be feasibly assigned to with respect to the h 's capacity and its own due dates is $[l_h, r_h[$. Now, the analogy to $\bar{\mathcal{I}}$ is readily available. Cliques of intervals of size at most 3 correspond to loads of vehicles of up to three shipments that can be feasibly assigned to the same vehicle. Due to $I = \{0, \dots, r_{|V|}\}$ and our construction of $\bar{\mathcal{I}}$ each subset of shipments that can be feasibly assigned to the same vehicle can be assigned to at least $|V|$ vehicles and, hence, there is no shortage of vehicles. The total occupation time of a vehicle which has assigned subset S' of shipments equals $\max\{p_{h_s} \mid s \in S'\}$ if $S' \neq \emptyset$ and zero otherwise. Hence, its total occupation time directly corresponds to the maximum weight among intervals in the corresponding clique. Therefore, a partition of second shipments into subsets assigned to the same vehicles such that total occupation time of these vehicles is O corresponds to a solution to $\bar{\mathcal{I}}$ with total maximum weight in cliques of O .

Now, it is easy to see that if there is a solution to $\bar{\mathcal{I}}$ with total maximum weight in cliques of O , we can construct a feasible solution to $\mathcal{I}'$ with total occupation time of $\lceil |V|/2 \rceil \cdot (p_{|V|} + 1) + O$ using the structure outlined above. $\square$

Thus, we can see that CTSP is a challenging optimization problem, where multiple issues trigger that it is notoriously hard to solve.

4 Mixed integer programs

4.1 Formulation using explicit arrival time variables

Applying the notation summarized in Table 1, this section presents a first (yet non-trivial) MIP model for CTSP that is solvable by an off-the-shelf MIP solver. This model, which we dub CTSP-MIP-1, consists of objective (1) subject to constraints (2) to (20).

CTSP-MIP-1:

$$\min Z(\vec{x}, \vec{y}, \vec{z}, \vec{t}, \vec{r}, \vec{g}) = \sum_{i \in I} r_i \quad (1)$$

H	Set of hubs (index $h \in \{0, 1, \dots, H \}$ with 0 being the depot)
S	Set of shipments (index $s \in S$)
I	Set of departure slots (index $i \in I$)
h_s	Destination hub of shipment $s \in S$
d_s	Due date of shipment $s \in S$
C^I	Maximum number of shuttles that are allowed to be hired
C_h^H	Storage capacity of hub h
C^V	Load capacity of shuttles
$x_{s,i}$	Binary variable: 1, if shipment $s \in S$ is assigned to departure slot $i \in I$; 0, otherwise
y_i	Binary variable: 1, if at least one shipment is assigned to departure slot $i \in I$; 0, otherwise
$z_{s',s}$	Binary variable: 1, if shipment $s' \in S$ has arrived at $h_{s'}$ up to d_s , $s \in S, h_s = h_{s'}$; 0, otherwise
$t_{i,h}$	Continuous variable: start time of the shuttle in departure slot $i \in I$ from $h \in H \cup \{0\}$
r_i	Continuous variable: occupation time of the shuttle in departure slot $i \in I$
g_s	Continuous variable: delivery time of shipment $s \in S$

Table 1: Notation

subject to:

$$\sum_{i \in I} x_{s,i} = 1 \quad \forall s \in S \quad (2)$$

$$\sum_{s \in S} c_s \cdot x_{s,i} \leq C^V \cdot y_i \quad \forall i \in I \quad (3)$$

$$\sum_{i \in I} y_i \leq C^I \quad (4)$$

$$t_{i,h} \geq t_{i,h-1} + T \cdot (p_h - p_{h-1}) \quad \forall i \in H, h \in H \setminus \{0\} \quad (5)$$

$$t_{i,h_s} \geq t_{i,h_s-1} + T \cdot (p_{h_s} - p_{h_s-1}) + P \cdot x_{s,i} \quad \forall i \in I, s \in S \quad (6)$$

$$t_{i,h} \leq t_{i,h-1} + T \cdot (p_h - p_{h-1}) + P \cdot \sum_{s \in S, h_s=h} x_{s,i} \quad \forall i \in I, h \in H \setminus \{0\} \quad (7)$$

$$t_{i,h_s} \leq t_{i,h_s-1} + T \cdot (p_{h_s} - p_{h_s-1}) + P \quad \forall i \in I, s \in S \quad (8)$$

$$g_s \geq t_{i,h_s} - M \cdot (1 - x_{s,i}) \quad \forall i \in I, s \in S \quad (9)$$

$$g_s \leq t_{i,h_s} + M \cdot (1 - x_{s,i}) \quad \forall i \in I, s \in S \quad (10)$$

$$g_s \leq d_s \quad \forall i \in I, s \in S \quad (11)$$

$$d_s + \epsilon - g_{s'} \leq M \cdot z_{s',s} \quad \forall s', s \in S, h_s = h_{s'} \quad (12)$$

$$\sum_{s' \in S, h_s=h_{s'}, d_{s'} \geq d_s} c_{s'} \cdot z_{s',s} \leq C_{h_s}^H \quad \forall s \in S \quad (13)$$

$$r_i \geq t_{i,h_s} - i - M \cdot (1 - x_{s,i}) \quad \forall i \in I, s \in S \quad (14)$$

$$x_{s,i} \in \{0, 1\} \quad \forall i \in I, s \in S \quad (15)$$

$$y_i \in \{0, 1\} \quad \forall i \in I \quad (16)$$

$$z_{s',s} \in \{0, 1\} \quad \forall s', s \in S, h_s = h_{s'} \quad (17)$$

$$t_{i,h} \geq i \quad \forall i \in I, h \in H \quad (18)$$

$$g_s \geq 0 \quad \forall s \in S \quad (19)$$

$$r_i \geq 0 \quad \forall i \in I \quad (20)$$

Objective function (1) minimizes the shuttles' total occupation time. Constraints (2) to (4) make sure that each shipment is delivered, no shuttle's load capacity is exceeded, and no more than C^I shuttles are used. Constraints (5) and (6) together with (18) bound start times of shuttles at hubs from below depending on whether or not a shipment has been delivered to the hub. Constraints (7) and (8) bound start times of shuttles at hubs from above depending on whether or not the shuttle visits a hub. Note that constraints (5) to (8) enforce $t_{i,h_s} = t_{i,h_{s-1}} + T \cdot (p_h - p_{h-1})$ if i does not visit h_s and $t_{i,h_s} = t_{i,h_{s-1}} + T \cdot (p_h - p_{h-1}) + P$ otherwise. Constraints (9) and (10) fix the delivery time t_s of $s \in S$ to the time the corresponding shuttle leaves destination hub h_s of s , while (11) ensures that no due date is violated. A trivial choice for big integer M is the start time of the last vehicle $\max\{i \mid i \in I\}$ at the last hub $|H|$ assuming that it visits each hub. Constraints (12) sets $\vec{z}$ according to delivery times. Note that we can choose ϵ appropriately such that $d_s + \epsilon$ lies properly between d_s and the earliest potential start time of a shuttle at h_s after d_s . The shuttle in each departure slot $i \in I$ gives rise to one such potential start time as $i + T \cdot p_{h_s} + k \cdot P$ with $k = \lfloor (d_s - i - T \cdot p_{h_s}) / P \rfloor + 1$. Constraints (13) ensure that hub capacities are not violated. Recall that we can restrict ourselves to points of time coinciding with the due date of at least one shipment s and that the capacity load at that time can be determined as the total size of shipments that have arrived up to this point and whose due date is at least d_s . Variables r_i are bounded from below in constraints (14), which allows to employ them in objective function (1) in order to evaluate the total shuttle occupation time. Finally, (15) to (20) specify the variables' domains.

CTSP-MIP-1 employs $\mathcal{O}(|S| \cdot (|I| + |S|))$ binary variables and has $\mathcal{O}(|H| \cdot (|H| + |I|) + |S| \cdot (|S| + |I|))$ constraints.

4.2 Formulation using implicit arrival times

Unfortunately, CTSP-MIP-1 contains several Big-M restrictions, which result in a weak linear relaxation. In this section, an alternative formulation (dubbed CTSP-MIP-2) is proposed that relies on cover inequalities to rule out infeasible assignments based on temporal or spacial incompatibilities. For this model, an extra variable $w_{i,h}$, controlling whether the shuttle in departure slot i delivers a shipment to hub h , is necessary. A summary of the newly added variable and parameters is given in Table 2.

CTSP-MIP-2:

$$\min Z(\vec{x}, \vec{y}, \vec{z}, \vec{w}, \vec{r}) = \sum_{i \in I} r_i \quad (21)$$

$w_{i,h}$	Binary variable: 1, if the shuttle in departure slot $i \in I$ delivers a shipment to hub $h \in H$; 0, otherwise
$\hat{H}_{s,j}$	Set of subsets of hubs $1, \dots, h_s - 1$ with cardinality j
$\bar{H}$	An element in $\hat{H}_{s,j}$

Table 2: Extended notation for second MIP model CTSP-MIP-2

subject to:

$$\sum_{i \in I} x_{s,i} = 1 \quad \forall s \in S \quad (22)$$

$$\sum_{s \in S} c_s \cdot x_{s,i} \leq C^V \cdot y_i \quad \forall i \in I \quad (23)$$

$$\sum_{i \in I} y_i \leq C^I \quad (24)$$

$$w_{i,h} \leq y_i \quad \forall i \in I, h \in H \quad (25)$$

$$w_{i,h_s} \geq x_{s,i} \quad \forall i \in I, s \in S \quad (26)$$

$$w_{i,h} \leq \sum_{s \in S: h_s = h} x_{s,i} \quad \forall i \in I, h \in H \quad (27)$$

$$r_i \geq T \cdot p_h \cdot w_{i,h} + \sum_{h'=1}^h P \cdot w_{i,h'} \quad \forall i \in I, h \in H \quad (28)$$

$$x_{s,i} = 0 \quad \forall i \in I, s \in S : d_s < i + P + T \cdot p_{h_s} \quad (29)$$

$$x_{s,i} + \sum_{h \in \bar{H}} w_{i,h} \leq j$$

$$\forall s \in S, i \in I, d_s \geq i + T \cdot p_{h_s} + P, \bar{H} \in \hat{H}_{s,j}, j = \left\lfloor \frac{(d_s - i - T \cdot p_{h_s} - P)}{P} \right\rfloor + 1 \quad (30)$$

$$z_{s',s} \leq 1 - x_{s',i} \quad \forall s', s \in S, i \in I, h_s = h_{s'}, d_{s'} \geq d_s, i + T \cdot p_{h'_s} + P > d_s \quad (31)$$

$$z_{s',s} \geq x_{s',i} \quad \forall s', s \in S, i \in I, h_s = h_{s'}, d_{s'} \geq d_s, i + T \cdot p_{h'_s} + h_{s'} \cdot P \leq d_s \quad (32)$$

$$z_{s',s} \geq x_{s',i} - \frac{1}{\left\lfloor \frac{(d_s - i - T \cdot p_{h_{s'}} - P)}{P} \right\rfloor + 1} \sum_{h'=1}^{h_s-1} w_{i,h'} \quad \forall s', s \in S, i \in I, h_s = h_{s'}, \quad (33)$$

$$d_{s'} \geq d_s, i + T \cdot p_{h'_s} + P \leq d_s < i + T \cdot p_{h'_s} + h_{s'} \cdot P$$

$$\sum_{s' \in S, h_s = h_{s'}} c_{s'} \cdot z_{s',s} \leq C_{h_s}^H \quad \forall s \in S \quad (34)$$

$$x_{s,i} \in \{0, 1\} \quad \forall i \in I, s \in S \quad (35)$$

$$y_i \in \{0, 1\} \quad \forall i \in I \quad (36)$$

$$z_{s',s} \in \{0, 1\} \quad \forall s', s \in S, h_s = h_{s'} \quad (37)$$

$$w_{i,h} \in \{0, 1\} \quad \forall i \in I, h \in H \quad (38)$$

$$r_i \geq 0 \quad \forall i \in I \quad (39)$$

CTSP-MIP-2 has the same objective function (21) as before. The core of the reformulation, coming by without an explicit calculation of the arrival time of each shipment, is presented in constraints (22) to (28). Equalities (22) assure that all shipments are assigned to shuttles, whose capacities are controlled by constraints (23). The number of shuttles is limited by (24). Three logic constraints define variable $w_{i,h}$. Constraints (25) enforce that a hub can only be visited by a shuttle if this shuttle is used. (26) assure that a hub is visited by a shuttle if a shipment is delivered there. Finally, (27) allow a hub visit only if a shipment is actually delivered in the respective hub. These constraints cut off solutions with artificial waiting time added to the delivery. The shuttle occupation is given by (28), which only depend on the hubs visited by each shuttle.

The remaining constraints regarding due dates and hub capacities are result of the cuts modeling the temporal and spacial incompatibilities of shipments. For the due date restriction, all pairs of shipments-to-slots assignments can be divided into three groups. The first group refers to shipments s sent in slot i that will always arrive late. This happens when due date d_s is larger than the arrival time of vehicle i , if it travels directly to the corresponding hub h_s . For this case, these assignments are infeasible and the corresponding variables can be set to 0, as shown in (29). If a vehicle is certain to arrive before the due date of the shipment, no restriction need to be applied to the assignment variables. This is the case when $d_s < i + T \cdot p_{h_s} + P \cdot h_s$. The third group contains the cases in which the number of visited hubs defines whether a shipment arrives before or after the due date. Assuming $i + T \cdot p_{h_s} + P \leq d_s < i + T \cdot p_{h_s} + P \cdot h_s$, there is a value j of hubs visits up to hub h_s for which the arrival time of the shuttle in departure slot i is larger than the due date. Value j is given by $\lfloor \frac{(d_s - i - T \cdot p_{h_s} - P)}{P} \rfloor + 1$ and is used in (30) to implement such a restriction. Constraints (30) use set $\hat{H}_{s,j}$, which contains all subsets of hubs before h_s with cardinality j . If j hubs are visited before h_s , shipment s cannot be assigned to this slot. If any of the hubs is not visited, variable $x_{s,i}$ is free to take any value.

The second reformulated constraint deals with the hub capacity. Variable $z_{s',s}$ models whether shipment s' uses hub capacity in hub h'_s at the due date d_s of shipment s . The modeling of such variables depends on three cases: Constraints (31) assure that a shipment s' assigned to the shuttle in departure slot i does not use space in a hub at the due date of shipment s , if the shuttle cannot arrive before d_s . If the shuttle in departure slot i arrives before due date s even by stopping at every hub it passes before h_s , a shipment s' must require hub space whenever being assigned to this shuttle according to (32). The third logical constraint refers to the case in which the number of visited hubs defines whether the shuttle arrives before or after a due date, see (33). Constraints (33) enforce $z_{s',s}$ to take value 1, if $x_{s',i} = 1$ and less than $\lfloor \frac{(d_s - i - T \cdot p_{h_{s'}} - P)}{P} \rfloor + 1$ hubs are visited before h_s . If more hubs are visited by the shuttle before h_s , the shipment arrives after due date d_s and variable $z_{s',s}$ is not forced to assume a non-zero value. This variable is used in (34) to account for hub capacities. Finally, (35) to (39) represent the variables' domain.

CTSP-MIP-2 employs $\mathcal{O}(|S| \cdot (|I| + |S|) + |I| \cdot |H|)$ binary variables and has $\mathcal{O}(|I| \cdot (|H| + |S|^2) + |S| \cdot 2^{|H|})$ constraints. Hence, we have significantly more binary variables and constraints in CTSP-MIP-2 than in CTSP-MIP-1. In particular, the number of constraints in CTSP-MIP-2 is exponential in $|H|$.

5 Computational study

This section is dedicated to our computational study. First, Section 5.1 elaborates our instance generator. Then, Section 5.3 explores the importance of the new problem features of CTSP. Afterward, Section 5.2 benchmarks the computational performance of our three solution approaches. Finally, we address managerial issues, by providing a sensitivity analysis to identify the main levers that influence the capacity utilization of the tunnel (Section 5.4) and by investigating the effects of demand pooling (see Section 5.5).

5.1 Instances generation

The dataset produced by our instance generator contains instances with 20, 40, and 60 shipments. Note that up to 60 shipments (e.g., EURO pallets) represent a substantial amount of cargo in the CTSP context because each shipment unites multiple actual shipments. A shipment refers to all customer orders serviced from the same micro hub on the same delivery tour. Hence, our 60 shipments must be multiplied by the capacity of the delivery vehicle (e.g., a cargo bike) to calculate (an upper bound on) the actual amount of cargo. All instances consist of a depot and four inner-city hubs ($|H| = 4$) in a tunnel with a length of $L = 100$ units. The depot is located at position 0 and the last hub at the end of the tunnel. The location of the other three hubs is determined by a random integer drawn from interval $[1,30]$, $[31,60]$, and $[61,90]$, respectively. The speed of the vehicles ($1/T$) is normalized to one length unit per time unit, and their capacities (C^V) are fixed to 50 units. The size of the shipments are integers drawn from a uniform distribution from 1 to 24. Given these limits, an average of four shipments fit into each vehicle.

The variable input parameters used to generate our dataset are the number of available departure slots, the unloading time, the hub capacity, the tightness of the due dates, and the group sizes of shipments with joint due dates. For each of these parameters, three values are selected that each hold for one third of the instances. Regarding the number of available departure slots, instances can either have limited potential time slots ($|I| = \lceil |S|/3 \rceil$), a medium number of options ($|I| = \lceil |S|/2 \rceil$), or ample possibilities ($|I| = |S|$). The time interval between slots (SI) also varies with this parameter. Starting from time 0, one time slot is considered for every 33 time units in the first configuration, every 20 time units for the medium one, and every 10 time units in the third configuration. The unloading time of shipments at the hubs can be either low ($P = 5$), medium ($P = 10$), or high ($P = 25$). The hub capacities are identical in all hubs and can also be

either low ($C^H = 50$), medium ($C^H = 75$), or high ($C^H = 100$). The due dates are drawn from a uniform distribution defined on interval $D = [T \cdot L + |H| \cdot P, SI \cdot (|I| + 1) + T \cdot L + |H|]$. For the instances with low due date tightness, the whole interval D is considered. For the medium values of tightness, only the last 85% values of the interval are considered, while the due dates of the third group with tight due dates only consider the last 70% values of the interval. All the due dates are rounded to the nearest integer. Finally, we also build instances, in which shipments are grouped and must be delivered to the same hub at the same due date. The instances can have no groups, consist of pairs of shipments, or have all shipments in groups of 3. Note that the last shipments form a isolated shipment or a pair if $|S|$ is not divisible by 3. For our computational tests, no upper limit on the number of tunnel slots is imposed, so that minimizing the vehicle utilization is the main objective.

Table 3: Varied parameter values used to generate the dataset.

Characteristic	Value (Low)	Value (Medium)	Value (High)
$ I $	$ S /3$	$ S /2$	$ S $
P	5	10	20
C^H	50	75	100
Due date	D	$D \cdot [0.15, 1]$	$D \cdot [0.3, 1]$
Groups	1	2	3

The varied parameters of our instance generator are summarized in Table 3. One instance is generated for each combination of these parameter values, resulting in 243 instances within each instance class having either 20, 40, or 60 shipments. Instances are discarded and replaced if they turn out being infeasible.

For the computational experiments, our three solution methods were used to solve the entire dataset within a time limit of 3600 seconds for each single instance. The Dantzig-Wolfe (DW) decomposition was implemented in Visual Basic.NET 15.0, while both MIP models and the linear relaxation of the master problem of DW were solved using Gurobi 9.1. For the experiments, a computer with an Intel i7 8700K processor with 6 cores at 4.0 GHz and 32 GB of RAM was used.

5.2 Computational performance

This section benchmarks the computational performance of our three solution methods. Table 4 summarizes the results of our performance tests. In the table, instances are grouped based on the number of shipments $|S|$ and results are displayed for the first MIP (CTSP-MIP-1 of Section 4.1), the second MIP (CTSP-MIP-2 of Section 4.2), and the DW decomposition (see appendix). Column ‘Best’ averages the results of the approach that contributes the best result for each particular instance.

We present the average results in accordance with the following performance criteria: the proportion of instances with verified optimal solutions discovered (denoted as ‘Share Opt’), the mean

Table 4: Performance results of the three solution approaches for CTSP.

Performance factor	$ S $	Best	CTSP-MIP-1		CTSP-MIP-2		DW	
		Average	Average	# Wins	Average	# Wins	Average	# Wins
Share Opt	20	0.96	0.74		0.90		0.76	
	40	0.43	0.15		0.43		0	
	60	0.14	0.02		0.14		0	
Time	20	279.87	1136.87	1	493.27	185	1253.13	47
	40	2280.74	3135.33	0	2280.74	104	3600	0
	60*	3205.00	3542.39	0	3205.00	27	3600	0
Upper Bound	20	473.16	473.36	242	473.16	243	474.14	239
	40	859.47	862.79	195	859.57	242	885.28	70
	60*	1266.52	1285.08	102	1267.54	227	1306.92	57
Lower Bound	20	471.93	452.68	179	469.79	222	462.48	190
	40	823.15	604.75	36	762.86	134	786.35	110
	60*	1183.52	714.02	4	972.10	36	1155.18	179

solution time (referred to as 'Time'), the average objective value of the best solution found (termed as 'Upper Bound'), and the mean lower bound (referred to as 'Lower Bound'). In the subsequent paragraphs, we shall conduct a comparative analysis of the outcomes achieved by our three distinct solution approaches based on these performance criteria.

- *Share Opt*: This criterion reports the share of instances that are solved to proven optimality within 3600 seconds. It is not surprising that larger instances with more shipments are, on average, harder to solve; 96% of the small instances but only 14% of the instances with 60 shipments are solved to proven optimality. The results also show a clear superiority of CTSP-MIP-2, which outperforms CTSP-MIP-1 and DW regarding the share of optimal solutions for all instance sizes.
- *Time*: For the 'Time' criterion, average solution times for each approach are reported in the column 'Average'. Column '# Wins', instead, reports the number of instances solved by the respective approach within the least amount of time. Note that a win is only computed if the instance is solved to optimality within the time limit. The column 'Best' shows the average processing time of the fastest method for each individual solution. Note that the average for the instances with size 60 is not computed over all 243 instances. DW is not able to find a feasible solution within the time limit for six instances, while CTSP-MIP-1 fails for one instance. These instances are excluded from the average to allow a fair comparison between the methods. The rows containing incomplete results are marked with *. Regarding the average solution times, the results are similar to the Share Opt criterion. CTSP-MIP-2 returns the results fastest among all competitors, especially for the larger instances. Unfortunately, DW is not able to solve instances with more than 40 shipments to optimality. One particular reason are certainly the symmetries of the CTSP that are not handled efficiently enough by our branching strategy. A set of shipments can also be sent in earlier or later shuttles whenever the respective vehicle has enough unused capacity and the

due dates and hub capacities are unviolated. However, adding a cut to reduce symmetries by enforcing later (or earlier) assignments may result in infeasible solutions whenever more flexibility is required.

- *Upper bound:* The average objective value of best solution found is reported by the ‘Upper Bound’ criterion. Again, CTSP-MIP-2 presents the best results. Column ‘Best’, displaying the best of all three methods, indicates merely slightly better results than CTSP-MIP-2 alone. The number of wins are counted for multiple methods if they achieve the same solution. For the large instances, CTSP-MIP-1 and DW find some of the best solutions that CTSP-MIP-2 cannot achieve, but overall CTSP-MIP-2 clearly outperforms its competitors.
- *Lower bound:* Finally, the lower bounds are compared in the last rows of the table. Here, a justification for our DW decomposition reveals. Even though DW is not able to solve many of the instances, it yields on average the best lower bounds for the medium and large instances. The difference in the quality of the lower bound is even larger for the instances with $|S| = 60$, for which DW obtained almost all the best lower bounds.

All in all, these performance results suggest the following conclusion: Solving CTSP-MIP-2 with a default solver delivers the best results among our three competitors and is the method of choice that is able to solve CTSP instance with up to $|S| = 60$ shipments with good performance. CTSP-MIP-1 is clearly outperformed in all instances, and DW seems a suitable choice only when having to determine lower bounds for large instances.

Furthermore, we aim to identify the types of CTSP instances that are particularly hard to solve. To do so, Figure 3 relates the mean gap (i.e., the difference between upper bound and lower bound divided by the upper bound of the best solution per instance in percent) to the varied input parameters of our instance generation.

Figure 3 suggests the following findings regarding the impact of the parameters on the solution process. As expected, the main complexity driver is the number of shipments, whose increase expands the solution space significantly. Next to this, a higher number of available departure slots and a large hub capacity allow for several combinations of shipments in the available vehicles. Although this flexibility is helpful for finding good solutions, proving their optimality by ruling out all alternatives is much harder. As a result, these instances present a higher solution gap for the dataset. Grouping shipments with similar due dates reduces the search space, since the shipments must often be shipped together. Therefore, instances with large groups have smaller solution gaps. The unloading time and due date tightness show less influential on the solution times.

To further analyze why the performance of DW cannot compete with that of CTSP-MIP-2, Table 5 reports more detailed statistics. Specifically, we list the average number of explored nodes and LPs as well as the instances of the pricing problem solved per node. We filter the results by the five parameters used for instance generation. First, there is a clear trend that the ratios of pricing subproblems and LPs solved per node increase with more shipments $|S|$. Consequently, within the

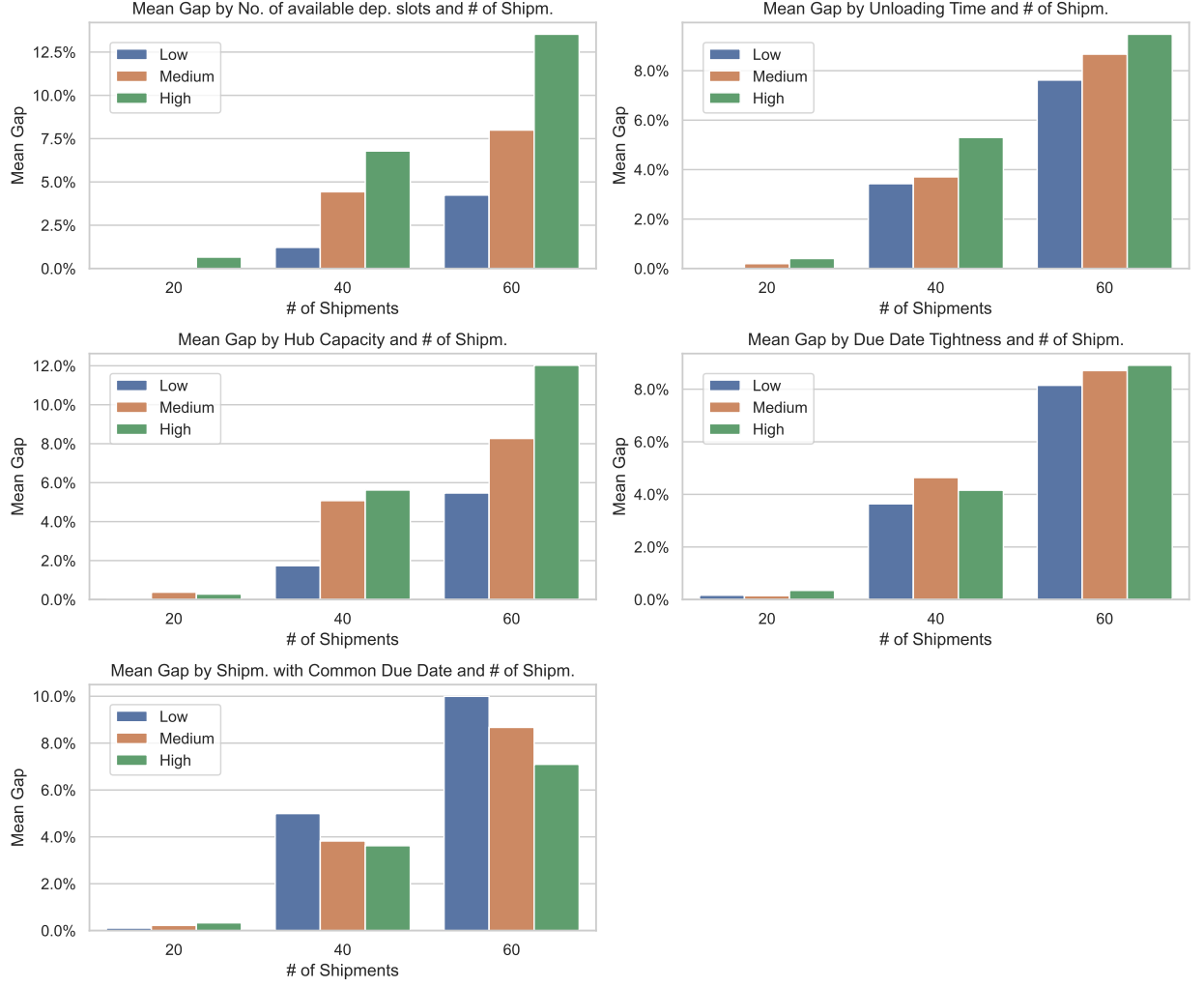


Figure 3: Plots with the average solution gap of the best solution found for each parameter and instance size.

3600-second time limit, fewer nodes are explored for the larger instances, which explains DW's problems with these instances. The only other parameter with significant impact is the number $|I|$ of available departure slots. For an extended number of available slots, the ratios of solved problems per node (mainly pricing subproblems) are higher, resulting in fewer nodes being explored within the time limit. Note that the average number of iterations to solve each node (LPs solved / node) is low: the average values per group vary between 1.06 and 1.75, meaning that often only one or two iterations per node are sufficient. This is roughly reflected also by the number of instances of the pricing problem solved. For a small, medium, and large numbers $|I|$ of slots ($|I| = \lceil |S|/3 \rceil$, $|I| = \lceil |S|/2 \rceil$, and $|I| = |S|$), we solve only a few more instances than one per available slot.

In Table 6, we present solution statistics for the small instances, separately for instances solved to proven optimality within the timeout and without proven optimal solution. Instances without optimality proof require the exploration of significantly more nodes, which is not always possible within the time limit. By comparing the ratios, we observe that the number of LPs solved per node

Table 5: Performance characteristics of the DW algorithm, reporting the number of nodes and ratios of iterations to solve the LP relaxation of the master problem and the pricing problems for the instances with feasible solution found within 3600 seconds.

	$ S $	# Nodes			Pricings solved / node			LPs solved / node		
		20	40	60	20	40	60	20	40	60
$ I $	L	9867	12366	6901	7.8	17.4	25.9	1.12	1.27	1.35
	M	15358	9900	4514	10.6	24.8	38.8	1.06	1.28	1.37
	H	10947	6466	2125	21.6	49.0	79.6	1.09	1.28	1.44
P	L	13270	10333	4490	13.5	27.6	40.2	1.08	1.26	1.37
	M	9462	10288	4360	13.6	25.2	38.3	1.13	1.25	1.40
	H	13547	8002	4171	12.5	29.7	41.5	1.06	1.33	1.33
C^H	L	7161	5403	3262	14.0	32.6	54.5	1.14	1.64	1.75
	M	14592	8969	4222	12.4	28.9	44.2	1.09	1.33	1.50
	H	14176	12248	4927	13.6	25.0	33.0	1.06	1.16	1.19
Due date	L	14161	10804	5292	13.9	27.5	37.5	1.08	1.27	1.34
	M	11171	9522	3872	12.5	26.5	43.6	1.09	1.27	1.44
	H	10898	8232	3825	13.0	28.0	39.3	1.10	1.29	1.35
Groups	L	11223	10089	4108	11.4	27.5	38.2	1.08	1.25	1.31
	M	12934	9439	5177	13.8	26.1	38.2	1.09	1.27	1.34
	H	11977	9078	3916	14.3	28.2	43.8	1.10	1.32	1.49

decreases slightly in several instance groups for the instances without optimality proof, whereas the number of instances of the pricing problem increases. The difference, however, is only observable in some of the groups. These numbers indicate that nodes in deeper levels require fewer iterations to be solved, but more pricing problem instances to enumerate all possible loading options.

After these performance tests, we turn to managerial issues in the next sections.

5.3 Importance of new problem features

This section explores the importance of our new CTSP problem features: non-zero unloading times, differently-sized shipments, and a more involved objective function. Recall that neglecting unloading times, standardizing shipments to unit loads, and employing a simplified objective function that merely minimizes the number of used shuttle slots allow [8] to solve the resulting simplified problem version of CTSP in polynomial time. Instead, our complexity analysis of Section 3.3 shows that including these problem features renders the CTSP strongly $\mathcal{NP}$ -hard. Hence, the price for the increase of modeling fidelity by including the new features is a more challenging optimization problem. This section explores whether the gains in modeling fidelity can justify the increased solution effort.

To do so, our first two experiments quantify how often neglected unloading times that actually exist and standardized unit-load shipments that actually vary in size lead to infeasibilities where considering the actual features allows to find a feasible solution. In a further experiment, we explore the impact of the objective function by comparing the simplified objective from previous research (i.e., minimizing the number of used shuttle slots) with our novel objective (i.e., minimizing shuttle

Table 6: Performance characteristics of the DW algorithm for the instances with 20 shipments separated by the instances with proven optimal solution (columns 'solved') and without optimality proof within 3600 seconds (columns 'not solved').

		# Nodes		Pricings solved / node		LPs solved / node		# Instances	
		solved	not solved	solved	not solved	solved	not solved	solved	not solved
$ I $	L	6047	39151	8.0	7.5	1.16	1.08	69	9
	M	7341	38206	10.6	10.5	1.08	1.06	57	20
	H	5444	22165	21.6	21.6	1.10	1.09	53	26
P	L	5479	33807	13.7	13.4	1.11	1.07	58	22
	M	5334	26234	12.2	14.8	1.17	1.10	65	16
	H	8208	31133	11.8	13.2	1.07	1.06	56	17
C^H	L	4722	25321	13.6	14.7	1.17	1.10	67	9
	M	7501	27507	10.0	13.6	1.09	1.09	51	28
	H	6972	38592	13.8	13.4	1.09	1.05	61	18
Due date	L	5376	35323	13.9	13.9	1.12	1.06	53	22
	M	6173	31475	11.0	13.7	1.11	1.07	65	16
	H	7180	24238	12.9	13.2	1.10	1.10	61	17
Groups	L	7321	35697	11.3	11.5	1.09	1.06	69	11
	M	4514	30785	13.1	14.0	1.14	1.07	53	25
	H	6663	27919	13.6	14.8	1.12	1.09	57	19

occupation time). To do so, we solve the CTSP with varying upper bounds on the number of used shuttle slots and compare the impact on shuttle occupation time. In the following, we describe the setups and results of these three experiments on the importance of the new problem features in detail.

- *Non-zero unloading times:* Neglecting unloading times cannot cause an instance to become infeasible. However, it may be the other way around. While the original instance is, in fact infeasible, we may obtain a feasible instance by setting unloading times to zero. To evaluate the price for neglecting unloading times, we proceed as follows. Among 100 instances produced with our instance generator and feasible with unloading time $P = 0$, we check the fraction of instances turning infeasible if the actual unloading times turn out as $P = 5$, $P = 10$, and $P = 20$ (while keeping all other parameter values). The result is that 46,5%, 47,4%, and 52,4% of all instances become infeasible. Hence, neglecting unloading times can be costly because schedules generated under this simplifying assumption are infeasible in (about) any second case in our experiments.
- *Unit-load shipments:* If an algorithm can only handle unit-load shipments, then a simple workaround is to standardize all differently-sized shipments to the maximum size among all shipments. This workaround guarantees that a schedule obtained for unit-load shipments remains feasible for the actual shipment sizes. On the negative side, the instance may become infeasible due to the overestimated shipment sizes. In our experiment, based on 100 instances, the latter problem occurs in almost all instances (except for a handful). Thus, especially if tunnel capacity is scarce, neglecting the actual shipment sizes can be a costly

simplification, obstructing the planning process.

- *Simplified objective:* An alternative goal for the CTSP involves minimizing the number of used shuttles, following the approach of Boysen et al. [8]. To gauge the significance of our extended objective, we contrast it with the simplified one by comparing the total shuttle utilization across solutions. For this experiment, the number of shuttles V^* used in the optimal (or best) solution of our CTSP (i.e., solved with a non-restrictive shuttle bound) constitutes the upper bound on the number of shuttles. Then, we test whether solutions using fewer shuttles are obtainable without constraint violations. This experiment solves all problem instances with a 3600-second time cap using the CTSP-MIP-2 model.

Our findings are compiled in Table 7. The table's left division records the count of instances that permit a feasible solution when fewer than the minimum number of shuttles are employed. Among the 243 instances evaluated per size, a significant fraction permits feasible solutions with one fewer shuttle slot. Saving even more shuttle slots is barely possible. Hence, our novel objective also supports reducing the number of used shuttle slots. Conversely, the table's right segment quantifies the average decrease in tunnel utilization attributed to a diminished number of shuttles, calculated exclusively for instances where a reduced shuttle fleet remains feasible. Hence, we conclude that, in a significant amount of instances, deviating from the minimum number of shuttles V^* can improve shuttle occupation. This potential of improvement gets lost when only considering the simplified objective.

Table 7: Comparison of alternative objective functions.

Instance size	# of Feasible Instances			Average Loss of Shuttle Occupation in %	
	$V^* - 1$	$V^* - 2$	$V^* - 3$	$V^* - 1$	$V^* - 2$
20	30	0	0	4.65	-
40	52	1	0	2.88	3.25
60	90	6	0	1.98	11.44

We conclude that the extended problem features of the CTSP considered in this paper can be valuable for real-world tunnel operations.

5.4 What drives tunnel utilization?

First, we aim to identify the main drivers that influence the utilization of the tunnel. In a sensitivity analysis, we relate the tunnel utilization to the parameters used for instance generation. Unfortunately, not every instance can be solved to proven optimality. Hence, the results in this section are based on the best solution of each instance found by all three solution approaches. To quantify the utilization of the tunnel, a performance measure called 'utilization rate' is introduced. The utilization rate of an instance is given by how loaded the vehicles are, weighted by the time

they use the tunnel in their forward movement. The utilization rate must therefore account for the load changes after each hub visit. A full vehicle has a high utilization at the start of the tunnel. Nonetheless, it may have a low overall utilization rate, if it delivers almost all shipments in the first hub and travels to the end of the tunnel with only one shipment. Moreover, the vehicle occupancy during unloading time is assumed to be the average of the load before and after unloading. In this way, we divide the actual utilization of each slot during its way through the tunnel by the available capacity of a fully loaded vehicle.

Figure 4 contains the plots of the average utilization rates of all instances classified by their size and one additional parameter (i.e., the number of available departure slots, the unloading time, hub capacity, due date tightness, and the group sizes of shipments with a common due date) of instance generation. These results suggest the following findings:

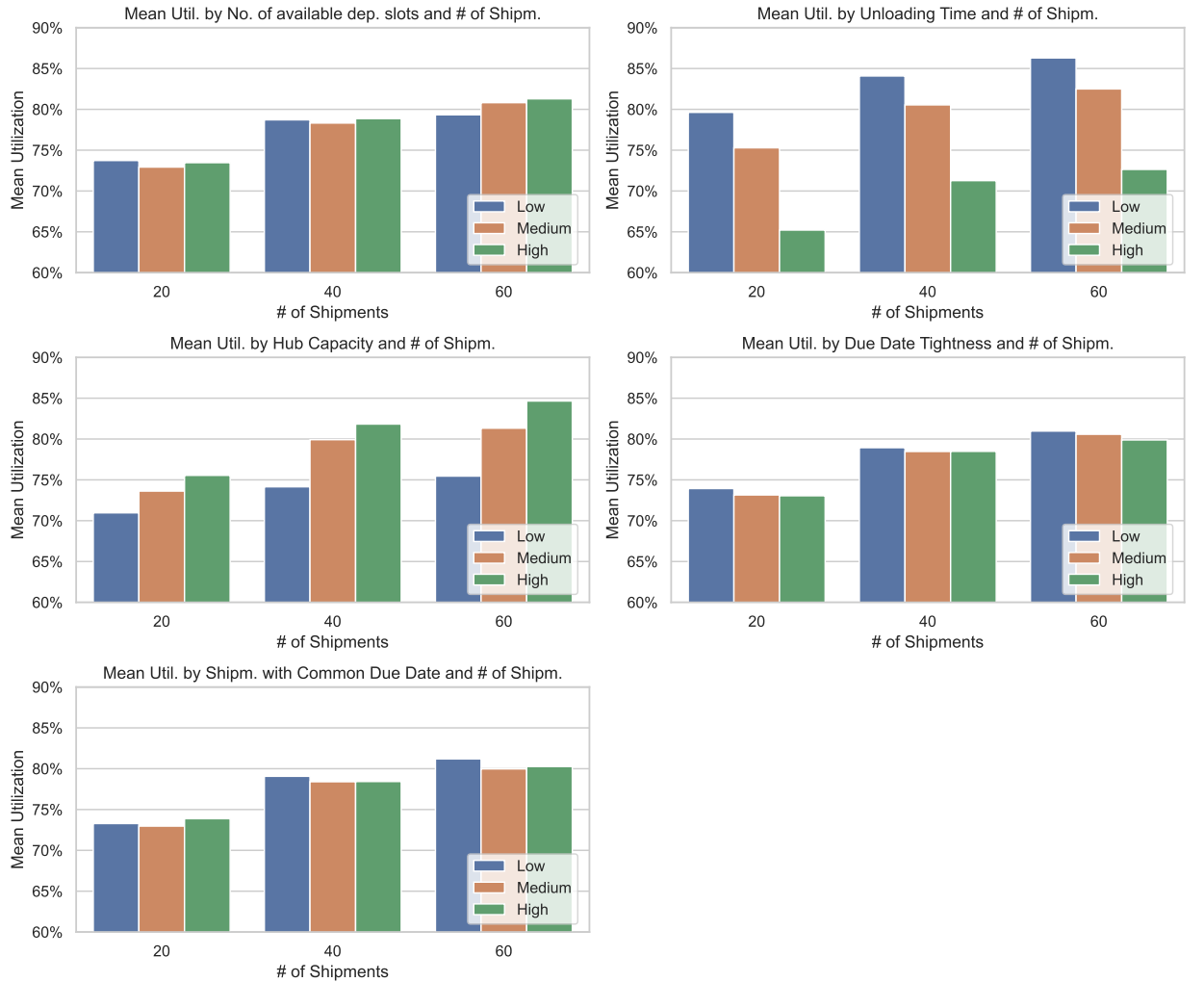


Figure 4: Plots with the average utilization rates of the best solution found for each parameter and instance size.

- *More shipments improve tunnel utilization:* First, we can observe an effect that is widely known for other matching tasks, e.g., when matching donated kidneys with patients [33]

or students with high schools [1]: More participants ease finding suitable matches. The same effect can be observed in our data, where the utilization rates vary from 73.4% for 20 shipments and 78.6% for 40 shipments to 80.4% for 60 shipments. More shipments to be transported by a shipper increase the chance that other shipments with similar due date and suitable size are available to share the departure slot of a vehicle. Thus, if the operator only services 'small' shippers with low demand for tunnel services, the risks of unused tunnel capacity increases.

- *Short unloading times and large hub capacities improve tunnel utilization:* The difference between a low and high unloading time results in a utilization rate decline from 83.3% to 69.7%. This shows the importance of fast handling systems to assure fast cargo unloading operations within the hubs (e.g., see the automated handling systems suggested by Magway . com). A small hub capacity can also impair the productivity of the system. A low hub capacity leads to an average utilization rate of 73.5%, while the high capacity configuration allows a utilization of 80.7%. We conclude that for an efficient tunnel-based cargo delivery system, we must invest not only in the tunnel itself (e.g., for boring) but also in automated handling systems and sufficient storage space to equip the hubs adequately.
- *Low impact of remaining parameters:* Neither the number of available departure slots, nor the tightness of due dates, nor the group sizes of shipments with common due dates show a significant impact on tunnel utilization.

Compared to the previous tunnel utilization results of Boysen et al. [8], based on unit-size shipments and no unloading times at hubs, our new results reveal a potential for even more unused tunnel capacity. Naturally, our less restrictive model, including individual shipment sizes and non-zero unloading times, increases the risk of half-empty shuttles.

Our analysis reveals that having more shipments that are scheduled jointly in a planning run strongly impacts tunnel utilization of the booked slots in a positive manner. A conclusion for the tunnel operator could thus be to only accept slot bookings from 'large' shippers with a lot of cargo to be transported through the tunnel. The consequence, however, is that the transportation demands of 'small' shippers are lost. A consolidation of shipments from multiple 'small' shippers that are jointly assigned to the available departure slots within mixed shuttles could be a feasible compromise. In this way, a high tunnel utilization while still servicing shippers with lower demand can be achieved. The resulting demand pooling effect is investigated in more detail in the following section.

5.5 On the demand pooling effect of shipment consolidation

When the operator of a cargo tunnel chooses not to reserve individual slots for each shipper but opts for shipment consolidation, it introduces the concept of demand pooling [e.g., see 10].

This concept holds the potential for a more efficient utilization of the allocated slots. However, there are drawbacks to consolidation, primarily manifesting as increased organizational complexity. Instead of explicitly assigning each slot's shipments to a specific shipper, an additional sorting process within the inner-city hubs becomes necessary to segregate cargo from multiple shippers transported in mixed shuttles. Consequently, this results in additional handling costs and an increased spatial requirement within the hubs. In this section, our focus is directed toward the positive aspect of this trade-off, where we evaluate whether shipment consolidation truly offers a substantial enhancement in tunnel utilization. Note that in line with Cattani and Schmidt [10], we assume a wide interpretation of demand pooling, not restricted to stochastic settings where the variability of the pool (and associated metrics such as safety stock) is lower than for the sum of the individuals. Here, we also take the well-known positive effect of more traffic on a matching platform [see 33, 1], in our case achieved by pooling the demand of multiple shippers, under this umbrella term.

To assess this effect, we set up the following computational experiment. Based on the 243 instances with 60 shipments, we simulate a booking process with and without shipment consolidation. For the latter, we partition the shipments equally among three shippers and process their booking sequentially. Thus, shipper 1 books first and is offered all available slots. After solving the resulting CTSP instance for the shipments of shipper 1, the slots of the solution are booked, so that shipper 2 has only those slot options available that are not already blocked by shipper 1. Shipper 3 can merely select among the slots not already booked by its two predecessors. The results of this process without shipment consolidation are compared with a unique optimization run of the respective CTSP instance for all 60 shipments. This emulates the case of shipment consolidation, where the operator registers the demand of all three shippers first and then plans the slot assignment jointly in mixed shuttles.

To implement this experiment, the hub capacity must also be accounted for. Constraints (13) are adapted to allow only the remaining capacity not yet used by the preceding shippers. This is done by adding an index s to parameter C_h^H . Due to the easiness of adapting the model, only CTSP-MIP-1 is fed into the default solver to generate the results of this experiment. The time limit for the computations is set to 3600 seconds.

Unfortunately, out of the 243 instances, only ten are feasible if no shipment consolidation is applied. The majority of the instances (i.e., 183) are already infeasible for the second shipper, because either due dates or capacities are violated. Another 50 instances are feasible for the first and second shipper, while the third shipper cannot find a feasible assignment for the last 20 shipments. The results for the only ten instances, which are solvable sequentially without consolidation, are displayed in Table 8. The values of the parameters used to generate these instances are given in columns 2 to 6. They represent the parameters defining the number of available slots ($|I|$), unloading time (P), hub capacity (C^H), due date tightness (Due Date), and group sizes of shipments with common due dates (Groups), respectively. Note that the number of available slots and the hub capacities are specially critical when no shipment consolidation is

applied. None of the instances with Low or Medium hub capacity is feasible, and a large number of available slots is also necessary for feasibility.

Table 8: Improvement effect of shipment consolidation.

Instance No.	$ I $	Parameter				Objective value		
		P	C^H	Due Date	Groups	Con.	No-Con.	Diff.
103	Medium	Low	High	Medium	Low	803	935	16.4%
105	Medium	Low	High	Medium	High	1009	1205	19.4%
129	Medium	Medium	High	Low	High	891	1031	15.7%
135	Medium	Medium	High	High	High	1023	1197	17.0%
182	High	Low	High	Low	Medium	978	1175	20.1%
209	High	Medium	High	Low	Medium	1074	1265	17.8%
210	High	Medium	High	Low	High	1139	1306	14.7%
211	High	Medium	High	Medium	Low	1230	1661	35.0%
215	High	Medium	High	High	Medium	1293	1467	13.5%
237	High	High	High	Low	High	1481	1779	20.1%

The large number of instances that are not solvable without shipment consolidation are a first indicator for the positive effect of shipment consolidation. Another one can be witnessed in Table 8, where we compare the objective values of the ten instances that are feasible with and without consolidation. The right part of Table 8 displays the objective function values with (column 'Con.') and without (column 'No-Con.') shipment consolidation. Column 'Diff.' lists the improvement of shipment consolidation over no consolidation. Over all ten instances, this improvement amounts to an average of 19.0% and a maximum improvement of 35%.

Therefore, we can infer that a tunnel operator should conduct a thorough assessment of the resources required for consolidating shipments of multiple shippers within inner-city hubs. If the resources necessary for this task can be managed within reasonable bounds, the practice of demand pooling, which enables the joint transportation of cargo from multiple shippers within mixed shuttles, holds the potential for significantly enhancing tunnel utilization.

6 Conclusions

This paper introduces three exact solution approaches for an extended version of the CTSP, where shipments must be assigned to shuttle vehicles that transport freight through an underground cargo tunnel while adhering to capacity and due date constraints. Instead of dispatching trucks into congested city centers for delivery, autonomous shuttles within tunnels offer a traffic-free means of transporting goods from peripheral depots to urban hubs. Nevertheless, the efficient utilization of the tunnel hinges significantly on how shipments are organized within the limited departure slots of tunnel vehicles and hub capacities.

Our problem setup differs from previous models in addressing the exact occupation time of shuttles in the objective function, non-zero unloading times at hubs, and individual shipment sizes. In particular, the latter feature adds to the complexity of our problem because finding good packing

strategies now becomes much more difficult. Furthermore, even with fully automated unloading processes at hubs, an unloading time of zero is unrealistic. Finally, tracking the exact shuttle occupation in the tunnel allows to reflect more realistic pricing schemes, not only including fixed cost per shuttle use but also a variable fee depending on the duration of use.

Our experiments identify two critical factors that can significantly enhance tunnel utilization for operators: (i) When assigning shipments to departure slots of shuttles, having a higher volume of shipments adds flexibility to find suitable matches for jointly utilized shuttles. Consequently, large shipping companies with substantial cargo volumes stand to benefit from cargo tunnel services. However, to accommodate smaller shippers, order consolidation, where shipments from multiple shippers share the same shuttles, proves to be an effective strategy for maintaining high tunnel utilization. (ii) Our analysis also reveals that slow handling times and limited storage capacities within inner-city hubs can impede capacity utilization within the cargo tunnel. Therefore, alongside the considerable tunnel construction costs, substantial investments in hub infrastructure, including automated shuttle unloading, are essential for the efficient operation of cargo tunnels.

Looking ahead, future research endeavors could challenge our exact solution methods. The extended version of the CTSP, involving non-zero handling times and individual shipment sizes, poses a formidable optimization problem. This challenge is underscored by our extensive efforts to streamline the Dantzig-Wolfe decomposition for the CTSP, without achieving superior results compared to a sophisticated MIP fed into an off-the-shelf-solver. Consequently, the development of more efficient exact but also heuristic solution methods for the CTSP represents an intriguing and demanding frontier for future research.

References

- [1] A. Abdulkadiroğlu, N. Agarwal, and P. A. Pathak. The welfare effects of coordinated assignment: Evidence from the New York City high school match. *American Economic Review*, 107(12):3635–3689, 2017.
- [2] P. Aellig. Cargo Sous Terrain. <https://www.cst.ch/> (last access: October 2023), 2020.
- [3] C. C. Anderson. If Elon Musk’s hyperloop sounds like something out of science fiction, that’s because it is. <https://www.businessinsider.com/elon-musks-hyperloop-science-fiction-2013-7?r=DE&IR=T#ixzz3Ws1tIWeb> (last access: October 2023), 2013.
- [4] D. Bliss. Mailrail: 70 years of automated underground freight transport. In *Compass Logistics Ltd, 2nd International Symposium on Underground Freight Transportation by Capsule Pipelines and Other Tube/Tunnel Systems: September*, pages 28–29, 2000.
- [5] N. Boysen, D. Briskorn, and S. Schwerdfeger. The identical-path truck platooning problem. *Transportation Research Part B: Methodological*, 109:26–39, 2018.

- [6] N. Boysen, S. Fedtke, and S. Schwerdfeger. Last-mile delivery concepts: A survey from an operational research perspective. *OR Spectrum*, 43:1–58, 2021.
- [7] N. Boysen, D. Briskorn, and J. Rupp. Optimization of two-echelon last-mile delivery via cargo tunnel and a delivery person. *Computers & Operations Research*, 151:106123, 2023.
- [8] N. Boysen, D. Briskorn, J. Rupp, and S. Schwerdfeger. Jam in the tunnel: On urban freight tunnels, their operational scheduling, and unused transport capacity. *Service Science*, 16(3):184–201, 2024.
- [9] F. Bruzzone, F. Cavallaro, and S. Nocera. The integration of passenger and freight transport for first-last mile operations. *Transport Policy*, 100:31–48, 2021.
- [10] K. Cattani and G. M. Schmidt. The pooling principle. *INFORMS Transactions on Education*, 5(2): 17–24, 2005.
- [11] F. Cavallaro and S. Nocera. Integration of passenger and freight transport: A concept-centric literature review. *Research in Transportation Business & Management*, page 100718, 2021.
- [12] E. Coffman, M. Garey, and D. Johnson. Approximation algorithms for bin packing: A survey. In *Approximation algorithms for NP-hard problems*, pages 46–93. Springer, 1996.
- [13] J. Cui and J. D. Nelson. Underground transport: An overview. *Tunnelling and Underground Space Technology*, 87:122–126, 2019.
- [14] J. Desrosiers and M. E. Lübbecke. A primer in column generation. In *Column generation*, pages 1–32. Springer, 2005.
- [15] V. Ghilas, E. Demir, and T. Van Woensel. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12–30, 2016.
- [16] V. Ghilas, E. Demir, and T. Van Woensel. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. *Transportation Research Part B: Methodological*, 91:34–51, 2016.
- [17] V. Ghilas, J.-F. Cordeau, E. Demir, and T. Van Woensel. Branch-and-price for the pickup and delivery problem with time windows and scheduled lines. *Transportation Science*, 52(5):1191–1210, 2018.
- [18] P. C. Gilmore and R. E. Gomory. Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. *Operations Research*, 12(5):655–679, 1964.
- [19] D. Hai, J. Xu, Z. Duan, and C. Chen. Effects of underground logistics system on urban freight traffic: A case study in Shanghai, China. *Journal of Cleaner Production*, 260:121019, 2020.
- [20] W. Hu, J. Dong, B.-g. Hwang, R. Ren, and Z. Chen. Hybrid optimization procedures applying for two-echelon urban underground logistics network planning: A case study of Beijing. *Computers & Industrial Engineering*, 144:106452, 2020.

- [21] W. Hu, J. Dong, B.-g. Hwang, R. Ren, and Z. Chen. A preliminary prototyping approach for emerging metro-based underground logistics systems: Operation mechanism and facility layout. *International Journal of Production Research*, 59(24):7516–7536, 2021.
- [22] HyperloopTT. HyperPort – Port of Hamburg Germany. <https://www.hyperlooptt.com/projects/hyperport/> (last access: October 2023), 2021.
- [23] T. Kirschstein. Planning of multi-product pipelines by economic lot scheduling models. *European Journal of Operational Research*, 264(1):327–339, 2018.
- [24] C. Kühnhold. Smart city loop. <https://www.smartcityloop.de/> (last access: October 2023), 2020.
- [25] T. V. Le, A. Stathopoulos, T. Van Woensel, and S. V. Ukkusuri. Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. *Transportation Research Part C: Emerging Technologies*, 103:83–103, 2019.
- [26] W. H. T. Meira, L. Magatão, S. Relvas, A. P. Barbosa-Póvoa, F. Neves, and L. V. Arruda. A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system. *European Journal of Operational Research*, 268(2):665–687, 2018.
- [27] B. Moffat. *Forty feet below: the story of Chicago's freight tunnels*. Interurban Press, 1982.
- [28] T. Nonner. Capacitated max-batching with interval graph compatibilities. *Theoretical Computer Science*, 613:79–93, 2016.
- [29] Y. Pan, C. Liang, and L. Dong. A two-stage model for an urban underground container transportation plan problem. *Computers & Industrial Engineering*, 138:106113, 2019.
- [30] PandoDaily. PandoMonthly: Fireside chat with Elon Musk. <https://www.youtube.com/watch?v=ueg0UmgKB4E> (last access: October 2023), 2013.
- [31] S. Powell, A. M. Campbell, and M. Hosseini. Underground freight transportation for package delivery in urban environments. *arXiv preprint arXiv:2405.04618*, 2024.
- [32] H. N. Psaraftis, M. M. Solomon, T. L. Magnanti, and T.-U. Kim. Routing and scheduling on a shoreline with release times. *Management Science*, 36(2):212–223, 1990.
- [33] A. E. Roth, T. Sönmez, and M. U. Ünver. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851, 2007.
- [34] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. In *Computer scheduling of public transport urban passenger vehicle and crew scheduling*, pages 269–280. North-Holland, Amsterdam, 1981.
- [35] M. Savelsbergh and T. Van Woensel. 50th anniversary invited article – City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

- [36] M. G. Speranza. Trends in transportation and logistics. *European Journal of Operational Research*, 264(3):830–836, 2018.
- [37] T. Standage. *The Victorian Internet: The remarkable story of the telegraph and the nineteenth century's online pioneers*. Phoenix London, 1998.
- [38] J. N. Tsitsiklis. Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22(3):263–282, 1992.
- [39] F. Vanderbeck. On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128, 2000.
- [40] B. Vernimmen, W. Dullaert, E. Geens, T. Notteboom, B. T’Jollyn, W. Van Gilzen, and W. Winkelmann. Underground logistics systems: a way to cope with growing internal container traffic in the port of Antwerp? *Transportation Planning and Technology*, 30(4):391–416, 2007.
- [41] J. G. Visser. The development of underground freight transport: An overview. *Tunnelling and Underground Space Technology*, 80:123–127, 2018.
- [42] H. Wei, A. Li, and N. Jia. Research on optimization and design of sustainable urban underground logistics network framework. *Sustainability*, 12(21):9147, 2020.
- [43] C. Wolmar. *The Subterranean Railway: How the London Underground was built and how it changed the city forever*. Atlantic Books Ltd, 2009.
- [44] X. Zhou, H. Zhang, R. Qiu, Y. Liang, G. Wu, C. Xiang, and X. Yan. A hybrid time MILP model for the pump scheduling of multi-product pipelines based on the rigorous description of the pipeline hydraulic loss changes. *Computers & Chemical Engineering*, 121:174–199, 2019.

A Dantzig-Wolfe decomposition

For the algorithm proposed in this appendix, the problem formulation is transformed by considering all possible sets of assignments of shipments to departure slots as variables. As there is a huge number of possible assignments, the Dantzig-Wolfe algorithm only considers a subset of possible assignments in the so-called restricted master problem [14]. Each of the assignments is called a column. The Dantzig-Wolfe algorithm is based on the well-known column generation paradigm, according to which columns (assignments) are added to the restricted master problem until the algorithm converges. In each iteration, the LP relaxation of the restricted master problem is solved and the dual variables are used to generate more columns [39]. The pricing problem, then, is to determine the column with minimal reduced cost. If it has negative cost, it is added to the restricted master problem. If it has reduced cost of zero, the current solution to the restricted master problem is optimal even for the full master problem (with all columns present). This column generation framework works for master problems with continuous variables. As the selection of a column (i.e., an assignment of a set of shipments to a slot) is a binary decision in our problem, a procedure is necessary to enforce the variables' integrality. The Dantzig-Wolfe algorithm integrates a branching scheme to solve the problem as a branch-and-price algorithm. The main ingredients of such an approach customized for our CTSP are elaborated in the following sections.

A.1 Restricted master problem

A column in our formulation of CTSP represents a selection of shipments assigned to a certain slot. This implies an arrival time for each of the assigned shipments (along with the information whether the selected shipments arrive before or after their due dates) and the duration of the delivery. In other words, a column k , representing the set S' of shipments to be assigned to slot i , implies the values $x_{s,i}^{(k)}$, $z_{s,s'}^{(k)}$, and $r^{(k)}$ of the corresponding variables used in Section 4 for each $s \in S'$, $s \in S$, and i . In the restricted master problem, variable $\lambda_k \in [0; 1]$ represents the portion to which column k is selected. Set K_i contains all columns representing assignments to departure slot i , and we let $K = \bigcup_{i \in I} K_i$.

The CTSP can then be reformulated using variable λ_k for each $k \in K$ with objective function (40) subject to constraints (41) to (46).

$$\min Z(\vec{\lambda}) = \sum_{k \in K} r^{(k)} \cdot \lambda_k \quad (40)$$

subject to:

$$\sum_{i \in I} \sum_{k \in K} x_{s,i}^{(k)} \cdot \lambda_k = 1 \quad \forall s \in S \quad (41)$$

$$\sum_{s' \in S, h_{s'}=h_s, d_{s'} \geq d_s} \sum_{k \in K} z_{s',s}^{(k)} \cdot c_{s'} \cdot \lambda_k \leq C_{h_s}^H \quad \forall s \in S \quad (42)$$

$$\sum_{k \in K} \lambda_k \leq \overline{C}^I \quad (43)$$

$$\sum_{k \in K} \lambda_k \geq \underline{C}^I \quad (44)$$

$$\sum_{k \in K_i} \lambda_k \leq 1 \quad \forall i \in I \quad (45)$$

$$\lambda_k \in \{0, 1\} \quad \forall k \in K \quad (46)$$

Constraints (41) assure that each shipment is assigned to a slot. Hub capacities are accounted for by constraints (42), while the number of shuttles is restricted by (43) and (44). Here, $\overline{C}^I$ and $\underline{C}^I$ represent an upper and lower bound on the number of employed shuttles, respectively. Initially, we have $\overline{C}^I = C^I$ and $\underline{C}^I = 0$, but these values can be adjusted in the branching process. As there is only one shuttle in each departure slot, constraints (45) assure that only one column can be selected for each slot. Finally, the variables' domains are set by (46).

A.2 Pricing problem

The pricing problem consists of finding a column with minimum reduced cost for the restricted master problem. For our CTSP, the pricing problem decomposes into independent problems depending on the slot i . That is, we find the best possible assignment of shipments to each slot individually. In the following, the MIP for the pricing problem is given in Section A.2.1 and a dynamic programming approach tackling this problem is proposed in Section A.2.2.

A.2.1 Mixed integer program for the pricing problem

A MIP for the pricing problem for departure slot i^* (fixed index) is given by objective function (47) subject to the constraints (48) to (59). In Table 9, the shadow prices of the restricted master problem are defined.

$\Pi_s^{Occ} \in \mathbb{R}$	Shadow price of the occurrence constraints (41) for each shipment s
$\Pi_s^{HC} \leq 0$	Shadow price of the hub capacity constraints (42) for each shipment s
$\Pi^{Vmax} \leq 0$	Shadow price of the maximal number of slots constraint (43)
$\Pi^{Vmin} \geq 0$	Shadow price of the minimal number of slots constraint (44)
$\Pi_{i^*}^{1C} \leq 0$	Shadow price of the slot constraints (45) for each slot i^*

Table 9: Additional notation for the pricing problem

$$\min Z_{i^*}(\vec{s}) = r_{i^*} - \sum_{s \in S} \Pi_s^{Occ} \cdot x_{i^*,s} - \sum_{s \in S} \sum_{s' \in S, h'_s = h_s} \Pi_{s'}^{HC} \cdot c_s \cdot z_{s,s'} - \Pi^{Vmax} - \Pi^{Vmin} - \Pi_{i^*}^{1C} \quad (47)$$

subject to:

$$g_s = i^* + T \cdot p_{h_s} \cdot x_{i^*,s} + \sum_{h \in H, p_h \leq p_{h_s}} P \cdot w_{i^*,h} \quad \forall s \in S \quad (48)$$

$$g_s \leq d_s + M \cdot (1 - x_{i^*,s}) \quad \forall s \in S \quad (49)$$

$$r_{i^*} \geq g_s - i^* \quad \forall s \in S \quad (50)$$

$$\sum_{s \in S} c_s \cdot x_{i^*,s} \leq C^V \quad (51)$$

$$w_{i^*,h_s} \geq x_{i^*,s} \quad \forall s \in S \quad (52)$$

$$w_{i^*,h} \leq \sum_{s \in S, h_s = h} x_{i^*,s} \quad \forall h \in H \quad (53)$$

$$g_s + M \cdot z_{s,s'} + M \cdot (1 - x_{i^*,s}) \geq d_{s'} + \epsilon \quad \forall s, s' \in S \quad (54)$$

$$x_{i^*,s} \in \{0, 1\} \quad \forall s \in S \quad (55)$$

$$z_{s',s} \in \{0, 1\} \quad \forall s', s \in S, h_s = h_{s'} \quad (56)$$

$$w_{i^*,h} \in \{0, 1\} \quad \forall h \in H \quad (57)$$

$$g_s \geq 0 \quad \forall s \in S \quad (58)$$

$$r_{i^*} \geq 0 \quad (59)$$

Objective function (47) minimizes the column's reduced cost, that is the occupation time minus the effects of the shadow prices of the restricted master problem. The delivery time of shipment s if assigned to i^* is defined by constraints (48). Note that g_s indeed implies the delivery time only, if $x_{i^*,s} = 1$ (and i^* plus the total waiting time at hubs otherwise). If $x_{i^*,s} = 1$, then the implied arrival time must not exceed the shipment's due date by constraints (49). The shuttle occupation time is then bounded from below in (50), as the difference of the shipments' delivery time and the departure time of the shuttle in departure slot i^* . Note that this bound is given by the total waiting time of i^* at hubs for shipment s not assigned to i^* . Constraint (51) ensures that the shuttle capacity is not violated. Constraints (52) take care that a shipment can only be delivered if the respective hub is visited by its shuttle, while Constraints (53) avoid stops in hubs without deliveries. Constraints (54) are logical restrictions controlling the link between variable $z_{s,s'}$, delivery time g_s , and due dates $d'_{s'}$. Finally, the variables' domains are set by (55) to (59).

A.2.2 A dynamic programming approach

The pricing problem shares similarities with a knapsack problem, since each shipment requires space in the shuttle and presents a value based on the shadow prices. The evaluation of the solutions, however, also depends on the shuttle occupation time and the shipments actual arrival times and, therefore, requires a tailored algorithm. As the pricing problem decomposes, the

dynamic programming (DP) approach described in this section solves the pricing problem for each slot i^* individually.

The DP proposed for the pricing problem is composed of two parts. First, each hub is considered individually. Then, the best assignments per hub are combined to consider the complete solution of the problem. Both steps are accomplished by DP approaches.

In the first part of the algorithm, the value $A(h, c, n)$ is calculated for each $h \in H$, $c \in \{0, \dots, C^V\}$, $n \in \{0, \dots, h-1\}$. $A(h, c, n)$ corresponds to the minimum contribution to the objective function of the pricing problem (corresponding to a fixed i^*) considering only shipments heading to hub h , using at most a shuttle capacity of c , and with exactly n hubs visited before hub h . Let, furthermore, $\bar{A}(h, c, n)$ be the set of selected shipments corresponding to $A(h, c, n)$. Naturally, the computation of $A(h, c, n)$ and $\bar{A}(h, c, n)$ can be done for each hub h and each value of n independently and, therefore, we assume h and n to be fixed in the following.

The set $S(h, n)$ contains all shipments whose due dates are at least $Arr := i^* + T \cdot p_h + P \cdot (n+1)$ and, thus, can be feasibly assigned to slot i^* if it stops at n hubs before reaching p_h . In the following we assume these shipments to be numbered from 1 to $|S(h, n)|$.

In this phase, only the dual variables for occurrence (Π^{Occ}) and hub capacity (Π^{HC}) are accounted for, since the others are constant for a given slot i^* . We consider shipments in $S(h, n)$ one by one and decide whether to assign them to the slot or not. $Z(s, c)$ represents the minimum contribution after considering shipments $1, \dots, s$ and occupying a capacity of c . The computations are detailed by Algorithm 1.

```

 $Z(0, 0) = 0$ 
For  $c = 1$  to  $C^V$ :
     $Z(0, c) = \infty$ 
For each  $s = 1, \dots, |S(h, n)|$ :
    For  $c = 0$  to  $C^V$ :
        if  $c_s \geq c$  then:
             $Z(s, c) = \min\{Z(s-1, c-c_s) - \Pi_s^{Occ} - \sum_{s' \in S, h_{s'}=h, Arr \leq d_{s'}, d_{s'} < d_s} \Pi_{s'}^{HC} \cdot c_s, Z(s-1, c)\}$ 
        else:
             $Z(s, c) = Z(s-1, c)$ 
For  $c = 0$  to  $C^V$ :
     $A(h, c, n) = Z(|S(h, n)|, c)$ 

```

Algorithm 1: Computation of $A(h, c, n)$ for all c with given h , n , and i^*

First, evaluations of initial states are set appropriately. In the second phase, the other states are evaluated recursively using the Bellman function. Finally, the values of $A(h, c, n)$ are determined as the best objective value of a shuttle visiting up to hub h , using a capacity of at most c units of space, while having visited n hubs before arriving at hub h .

After determining $A(h, c, n)$ and $\bar{A}(h, c, n)$ we, in the second part, combine these sets of shipments to full loads of slots. Value $D(h, c, n)$ corresponds to the minimum reduced cost for slot i^* if $n+1$

hubs are visited, hub h is visited last, and a total capacity of at most c is occupied when leaving the depot. In Algorithm 2, the transport times and unloading times affecting the occupation time of i^* are considered explicitly, while dual variables are considered implicitly only since they are incorporated in value $A(h, c, n)$.

```

For each  $h = 1, \dots, |H|$ :
  For  $c = 0$  to  $C^V$ 
     $D(h, c, 0) = T \cdot p_h + P + A(h, c, 0)$ 
  For  $n = 1$  to  $h - 1$ 
    For  $c = 0$  to  $C^V$ 
       $D(h, c, n) = \min_{\substack{c' \in [0, c], \\ h' \in [n, h-1]}} \{T \cdot (p_h - p_{h'}) + P + A(h, c', n) + D(h', c - c', n - 1)\}$ 

```

Algorithm 2: Computation of $D(h, c, n)$ for all h, c , and n with given i^*

Value $D(h, c, 0)$ is initialized with $A(h, c, 0)$ plus the required amount of time $T \cdot p_h + P$ to deliver shipments to h . This timespan accounts for the portion of r_{i^*} due to delivering shipments to h as first hub. For determining value $D(h, c, n)$, $n > 0$, we add a single set of shipments to be delivered to a new last hub falling back recursively on values of $D(h', c', n - 1)$, $h' < h$ and $c' \leq c$, using the Bellman function. To account for the new hub, we add P and the extra travel time as the increment of r_{i^*} caused by adding the last set of shipments.

We can now determine the objective value of the column with minimum reduced cost (represented by the total load of i^* corresponding to $\min\{D(h, C^V, n) \mid n \in H, n = 0, \dots, |H| - 1\}$) as

$$\min\{D(h, C^V, n) \mid n \in H, n = 0, \dots, |H| - 1\} - \Pi^{Vmax} - \Pi^{Vmin} - \Pi_{i^*}^{1C}.$$

If there is a column with negative reduced cost, we add it to the master problem. If we obtain multiple such columns for different values of i^* , we add all of them. If there is no such column the current solution to the LP relaxation of the restricted master problem is found.

A.3 Branching scheme

Unfortunately, solving the LP relaxation of the restricted master problem and adding new columns until optimality is reached does not guarantee an integer solution. Therefore, we employ two branching schemes that partition the solution space by fixing the assignment of shipments to a departure slot or limiting the flexibility on how to combine two shipments in a slot.

The first branching scheme enforces that shipments are fully assigned to a single slot. Whenever this is not the case in a solution, we identify the largest shipment s in terms of shuttle capacity which is not assigned to a unique slot and the first slot i with $\sum_{i' \in I, i' \leq i} \sum_{k \in K_{i'}} \lambda_k \geq 0.5$, that is the first slot up to which half of s is assigned to. We, then, enforce s to either be assigned only to slots smaller or equal i or be assigned only to slots larger i (if i is the last slot with a non-zero value, then we use $i - 1$ instead of i). Note that this might remove s from $S(h, n)$ for pricing problem instances associated with some slots, but no further adaptations are required.

While the first branching scheme can enforce a unique slot for each shipment it cannot enforce unique loads of shipments for each slot. When we encounter such a solution, we follow the idea of Ryan and Foster [34] and consider two shipments s and s' which are in some loads together and in some not. If there are multiple such pairs, we choose the one which has joint assignments with a total weight closest to 0.5. We, then, enforce s and s' to either be in the same load or not to be in the same load. In the former case, we merge s and s' into a composite shipment accounting for the joint impact on reduced cost of a column, load of vehicle and hub capacities, and occupation time of a slot. The composite shipment gets assigned the earlier of both due dates. In the latter case, we consider two instances of pricing problems dropping s from $S(h, n)$ in one and dropping s' from $S(h, n)$ in the other. Note that this approach requires the potential solution of several pricing problem instances if multiple branching steps are required. Therefore, we only branch on shipments assigned to the same hub. In this way, only the first part of the DP approach (the calculation of $A(h, c, n)$) is performed multiple times. The second phase of the algorithm is performed only once using the best obtained values of $A(h, c, n)$.

The elaboration of the branching scheme completes our customized Dantzig-Wolfe approach for the CTSP.