Robust Continuous-Time Service Network Design under Travel Time Uncertainty

Shengnan Shu
Department of Logistics and Maritime Studies, Hong Kong Polytechnic University, shengnan.shu@connect.polyu.hk

Zhou Xu
Department of Logistics and Maritime Studies, Hong Kong Polytechnic University, lgtzx@polyu.edu.hk

Jin Qi
Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology, jinqi@ust.hk

The continuous-time service network design problem (CTSNDP) has wide applications in the field of transportation, but it is complicated by travel time uncertainty resulting from unpredictable traffic conditions. Incorporating uncertain travel times poses a significant challenge, as time-indexed mixed-integer linear programming (MILP) formulations commonly used to solve the CTSNDP with deterministic travel times become impractical. This is due to their inability to distinguish between decisions that rely on travel times and those that do not. To tackle this challenge, we study a robust CTSNDP under travel time uncertainty, aiming to design a transportation service network with reliable operational efficiency even in the presence of travel time deviations. To incorporate the travel time uncertainty in a tractable manner, we propose a novel consolidation-indexed MILP formulation for the deterministic CTSNDP, eliminating the requirement for time indices. This enables us to derive MILP formulations for both a robust optimization model and a robust satisficing model of the CTSNDP under travel time uncertainty. To solve these formulations exactly, we have developed two tailored column-and-constraint generation methods. Our computational results demonstrate the effectiveness of these solution methods and the tractability of the proposed formulations. Furthermore, the robustness of the solutions obtained has also been verified, and the trade-off between the robustness and its price has been highlighted.

Key words: service network design; continuous time; travel time uncertainty; robust optimization; robust satisficing; exact algorithm; column-and-constraint generation

History: October 27, 2023

1. Introduction

In the transportation industry, a significant portion of the freight is moved by consolidation carriers, including railroads, container shipping lines, less-than-truckload motor carriers, and regular and express postal service providers. These consolidation carriers transport shipments that are small compared to their vehicles' capacities. As a result, they need to consolidate their shipments to achieve cost-effectiveness, which poses a service network design problem (SNDP).

The SNDP involves the routing of shipments of different quantities from their origins to their destinations through a network of terminals, where shipments can be transferred from inbound vehicles to outbound vehicles. Each shipment has an available time for departure from its origin and a due time for arrival at its destination. To transport shipments between terminals, one or more vehicles with limited capacities need to be used, incurring fixed costs on a per-vehicle basis and flow costs on a per-shipment-quantity basis. At each terminal, when multiple shipments are consolidated, the outbound vehicle carrying these shipments cannot be dispatched until all the inbound vehicles bringing these shipments have arrived. Accordingly, a classic SNDP seeks to determine both the routing and the consolidation plans of shipments, as well as the numbers and dispatch times of vehicles on each terminal-to-terminal movement, so that the shipment available times and due times are met. Its objective is to minimize the total operational cost, including both the fixed costs and flow costs.

The classic SNDP and other variants of the SNDP have been extensively studied in the operations research literature since the 1990s (Crainic and Rousseau 1986, Farvolden and Powell 1994), due to their wide applications and theoretical significance. However, the existing studies primarily focus on deterministic variants of the SNDP, assuming that all problem parameters, such as shipment quantities and travel times, are known in advance. To model the *deterministic* variants of the SNDP, a widely used technique is *discretization*, which involves discretizing the planning horizon into a sufficient number of time points (see, for example, Pedersen et al. 2009, Andersen et al. 2009a,b, Wieberneit 2008, Andersen et al. 2011, Crainic et al. 2016). Using these time points, a deterministic SNDP can be modeled on a time-expanded network, which consists of time-space nodes and arcs connecting these nodes. Each time-space node represents a combination of a time point and a terminal, while each arc between two time points represents a shipment's movement between terminals or its waiting at a terminal during a specific period.

The time-expanded network constructed from the discretization can be effectively used to incorporate decisions involved in the SNDP. Specifically, one or more decision variables can be defined for each shipment and for each arc between two time-space nodes, which can then be used to represent shipments' routing and consolidations, as well as to determine the numbers of vehicles, their loads, and dispatch times for each terminal-to-terminal movement of the shipments. Based on this approach, a mixed-integer linear programming (MILP) formulation can be established for the SNDP. It is a *time-indexed* formulation, since the decision variables involved are indexed by a pair of time points of the time-space nodes. The time-indexed MILP formulation can be solved directly by commercial optimization solvers. It has received significant attention in the literature

due to its potential extension to include additional management issues, such as asset repositioning, multiple transportation modes, and transit time constraints (Andersen et al. 2009a, 2011, Crainic et al. 2016, Scherr et al. 2019, Chouman and Crainic 2021, Zhu et al. 2014, Hellsten et al. 2021).

It is worth noting that the time-indexed MILP formulation is typically an approximation of the SNDP, where the planning horizon is continuous and vehicles can be dispatched at any time. The selection of an appropriate level of time discretization poses a challenge, as it directly affects the quality of the obtained solutions. Achieving high-quality solutions often requires a fine discretization, which, in turn, leads to a large and typically intractable time-indexed MILP formulation of the SNDP.

To solve the continuous-time variant of the SNDP, or *CTSNDP* in short, Boland et al. (2017) proposed a Dynamic Discretization Discovery (DDD) algorithm, which was further enhanced and extended by Hewitt (2019), Marshall et al. (2021), and Shu et al. (2023). The DDD algorithm is an exact solution algorithm for the CTSNDP that utilizes an iterative process to adjust the level of discretization. Each iteration of this iterative process generates a new time-expanded network, resulting in a different time-indexed MILP formulation being solved. It has been proven that the solutions obtained through this iterative process converge to the optimal solution of the CTSNDP.

Note that the formulations and solution algorithms introduced above are applicable to only deterministic variants of the SNDP. In contrast, the focus of our study is on the development of an exact solution algorithm for a new variant of the CTSNDP that incorporates travel time uncertainty. This is an under-explored and very challenging task that holds significant practical value, as will be outlined in the following explanation.

1.1. Challenges to Incorporating Travel Time Uncertainty

The two primary sources of uncertainty encountered in practical applications of the SNDP are travel times and shipment demands. The majority of existing studies that incorporate uncertainty in the SNDP, however, focus solely on demand uncertainty. These studies typically use a stochastic optimization approach (Lium et al. 2009, Hoff et al. 2010, Bai et al. 2014) or a robust optimization approach (Wang and Qi 2020, Atamtürk and Zhang 2007, Koster et al. 2013) to optimize the service network design for either the average performance or the worst-case performance, respectively. When vehicles' dispatch times are involved, all existing solution methods utilize the time-indexed formulation of the deterministic SNDP to incorporate uncertain demands (Lium et al. 2009, Hoff et al. 2010, Bai et al. 2014, Wang and Qi 2020). This is achievable because, even with the demand uncertainty, all the decision variables to be optimized retain the same time indices as those in the time-indexed formulation of the deterministic SNDP, although they need to be classified into two

groups. The first group pertains to here-and-now decisions, such as the dispatch times of the vehicles, which need to be decided on before having knowledge of the actual shipment demands. The second group relates to wait-and-see decisions, such as the actual routing of the shipments, which are made only after the actual shipment demands are known. Accordingly, the time-indexed formulation of the deterministic SNDP can be directly extended to establish a two-stage optimization formulation for the SNDP under demand uncertainty. At the first stage, the here-and-now decisions are optimized, while at the second stage, which is also called the recourse stage, the wait-and-see decisions are optimized based on the given here-and-now decisions and the revealed uncertain demands. Such a two-stage optimization formulation is often computationally tractable, since there are several known exact solution methods that can be applied, including the Benders decomposition method (Thiele et al. 2009) and the column-and-constraint generation method (Zeng and Zhao 2013).

Despite its significance, travel time uncertainty has rarely been considered in the existing literature on SNDP. Unlike uncertain demands, incorporating uncertain travel times poses a significant challenge, since the time-indexed formulation of the deterministic SNDP becomes impractical. In particular, here-and-now decisions that do not rely on travel times, such as the numbers of vehicles required, should not be represented by time-indexed decision variables. For wait-and-see decisions that rely on travel times, although they could be represented by time-indexed decision variables, such variables would need to have different time indices for different realizations of the uncertain travel times. For example, the actual departure times of the vehicles are among such wait-and-see decisions, since no vehicle can depart until all its shipments arrive. As a result, the time-indexed formulation of the deterministic SNDP cannot be directly utilized to establish a tractable two-stage optimization formulation for the SNDP under travel time uncertainty. As we will show in this paper, it is therefore advantageous to derive a new formulation of the deterministic SNDP that does not involve time indices. By doing so, we can properly distinguish and represent here-and-now decisions and wait-and-see decisions to establish a tractable two-stage optimization formulation under travel time uncertainty.

In this paper, our focus is on studying the CTSNDP under travel time uncertainty, where the planning horizon is continuous. Due to the impracticality of the time-indexed formulation for the deterministic CTSNDP, exact solution methods like the DDD algorithm, which were developed for the deterministic CTSNDP, cannot be applied to the CTSNDP under travel time uncertainty. Moreover, while travel time uncertainty has been considered in many studies on other transportation problems, such as the vehicle routing problem with time window (Hu et al. 2018, Adulyasak and Jaillet 2016), shortest path problem (Chen et al. 2016, Hu et al. 2018), and traveling

salesman problem (Chassein and Goerigk 2016, Zhang et al. 2021), the CTSNDP under travel time uncertainty presents another unique challenge. Specifically, it must account for delays resulting from the synchronization of multiple shipments that require consolidation for transportation. When one of these shipments is delayed, it can cause delays for all the others waiting for consolidation at a terminal. These delays can then propagate and cause additional delays for shipments at other terminals. We refer to this phenomenon as *consolidation delay propagation*, which further complicates the formulation and solution of the CTSNDP under travel time uncertainty.

1.2. Related Work

Despite its significant importance, travel time uncertainty has rarely been considered in existing studies on the SNDP. Among the studies that do consider it, all have overlooked the consolidation delay propagation caused by travel time uncertainty and consolidations (see, for example, Yao et al. 2014, Zhao et al. 2018, Liang et al. 2019, Lanza et al. 2021), except for three studies (Demir et al. 2016, Hrušovskỳ et al. 2018, Layeb et al. 2018). These three studies focus on a restricted variant of the SNDP for intermodal transportation of energy-wares, in which transportation services for shipment movements can only be selected from a small candidate set given in advance. In these three studies, a stochastic optimization approach is adopted to incorporate travel time uncertainty and optimize the average performance of the service network design. However, this approach requires complete probability information about travel times, which is difficult to know in advance. Furthermore, solving an optimum solution to a stochastic optimization model derived from this approach can be very challenging, due to the vast number of decision variables and possible realizations of uncertainty factors. As a result, the three aforementioned studies either apply an approximation method based on limited samples of travel time realizations (Demir et al. 2016) or some simulation-based heuristic methods (Hrušovskỳ et al. 2018, Layeb et al. 2018).

To the best of our knowledge, our study is the first to adopt a robust optimization approach to study the CTSNDP under travel time uncertainty. Our goal is to design a transportation service network that maintains reliable operational efficiency even under travel time deviations. The robust optimization approach is known to only require a distribution-free uncertainty set that defines the possible realizations of uncertainty factors (see, for example, Bertsimas and Sim 2004, Ben-Tal et al. 2009, and Bertsimas et al. 2011, for general theories and applications of robust optimization). It relaxes the need for complete information about the probability of uncertain factors, and has an optimization formulation that often has a tractable reformulation that can be efficiently solved to an optimum. In the classic robust optimization approach, the objective is to optimize the worst-case objective value of a solution over different realizations of uncertain factors. A recent study by Long

et al. (2023) proposes a new approach, named the robust satisficing approach. This approach aims to ensure a solution that best achieves a prescribed target of the objective value with the worst-case normalized magnitude of the violation from the target being minimized. It has been demonstrated to have the advantage of improving out-of-sample performance over the classic approach. Several applications of the robust satisficing approach have been studied in operations management. Zhou et al. (2022) propose a resource satisficing index that extends the criterion adopted in the robust satisficing approach for the scheduling of patient admissions. Cui et al. (2023) apply the robust satisficing approach in solving a two-stage resource pooling problem where the objective function of the second-stage problem is supermodular and increasing. The robust CTSNDP studied in this paper differs from these problems in the literature, as it is a two-stage optimization problem with a second-stage objective function that is not necessarily supermodular. As a result, the solution methods developed for the above problems are not applicable to the robust CTSNDP.

As demonstrated in the existing literature (see, for example, Ben-Tal et al. 2004, Bertsimas et al. 2011, Simchi-Levi et al. 2019), many robust optimization problems can be formulated as a two-stage optimization model, including the SNDP with demand uncertainty (Wang and Qi 2020, Atamtürk and Zhang 2007). In such a two-stage model, the first stage involves making here-and-now decisions before uncertain factors are realized, while the second stage involves making wait-and-see decisions after the uncertain factors are realized. It is known to be less conservative than a single-stage model (Atamtürk and Zhang 2007).

To apply the robust optimization approach to the CTSNDP under travel time uncertainty, we must first incorporate travel time uncertainty into the CTSNDP formulation. As discussed in Section 1.1, this is a challenging task, as the time-indexed formulations commonly used for the deterministic CTSNDP become impractical. The MILP formulations presented in Demir et al. (2016) and Hrušovskỳ et al. (2018) for the SNDP in intermodal transportation of energy commodities consist of decision variables with service indices, which are essentially time-indexed, as services are defined by the departure and arrival terminals and times of the vehicles. In a recent study, Hewitt and Lehuédé (2023) propose a new MILP formulation for the deterministic CTSNDP, which eliminates the time indices in the decision variables. However, this new formulation requires a set of all possible shipment combinations for consolidations. The number of decisions and constraints is proportional to the number of shipment combinations, which can grow exponentially with the number of shipments. As a result, this formulation can be very challenging to solve optimally for the deterministic CTSNDP, let alone its extensions for problem variants with travel time uncertainty. In contrast, this study presents a new compact formulation of the deterministic CTSNDP without time indices, which can be extended to incorporate travel time uncertainties while ensuring the tractability of the solution.

1.3. Contributions of This Paper

In this study, we tackle the challenge of formulating and solving the robust CTSNDP under travel time uncertainty. We first propose a novel MILP formulation for the deterministic CTSNDP that eliminates the need for time indices. We then extend this formulation to derive a classic robust optimization model and a robust satisficing model to incorporate travel time uncertainty. Both of these two models involve two stages of optimization. In the first stage, here-and-now decisions, including routing and consolidation of shipments, are optimized before actual travel times are realized. In the second stage, wait-and-see decisions, including the actual departure schedules of the vehicles, are optimized after the actual travel times are realized. Our models, being the first of their kind in the literature, are based on polyhedral uncertainty sets. Their extensions to distributional robustness and data-driven robustness will be explored in future research.

To solve the newly derived robust optimization model and robust satisficing model, we develop two exact algorithms, respectively. Both algorithms follow a column-and-constraint generation (C&CG) framework proposed by Zeng and Zhao (2013). This framework has been widely used to solve two-stage robust optimization models for various problems (see, for example, Zeng and Zhao 2013, Wang and Qi 2020), but has rarely been applied to two-stage robust satisficing models. The critical step in our C&CG algorithms is its solution to a subproblem that finds the worst-case realized travel times for any given first-stage solution. For our robust optimization model, we can formulate this subproblem as a mixed-integer linear program and solve it directly by an optimization solver. However, for our robust satisficing model, the complexity of the subproblem increases, necessitating our development of a novel bisection search procedure to solve it. Through extensive computational experiments conducted on randomly generated instances of the CTSNDP under travel time uncertainty, we have successfully demonstrated the tractability of our proposed models and the effectiveness of our developed algorithms.

The major contributions of our study are summarized as follows:

- We have developed a new MILP formulation for the deterministic CTSNDP that does not involve any time indices. Instead, this new formulation is defined by decision variables and constraints with indices associated with shipment consolidations, being referred to as a *consolidation-indexed* formulation. This not only enables us to derive tractable formulations of the robust CTSNDP with travel time uncertainty, but also opens a new direction for future study on the deterministic CTSNDP.
- We have derived a novel robust optimization model and a novel robust satisficing model for the robust CTSNDP under travel time uncertainty, based on polyhedral uncertainty sets. To the

best of our knowledge, these are the first such models of their kind. They have thus established a strong modeling foundation for future research in this area.

- We have developed two tailored C&CG algorithms to solve the newly derived robust optimization and robust satisficing models, respectively. To the best of our knowledge, these are also the first such exact algorithms for the CTSNDP under travel time uncertainty. Moreover, our paper is the first study that demonstrates the success of C&CG algorithms in solving two-stage robust satisficing models with a polyhedral uncertainty set. The convergence guarantee of this C&CG algorithm relies on a new bisection search procedure. These new algorithms have great potential for future extensions to other optimization problems that incorporate uncertainties.
- We have conducted extensive computational experiments over randomly generated instances to access the proposed optimization models and solution algorithms for the robust CTSNDP under travel time uncertainty, demonstrating their practical usefulness. The computational results also confirm the robustness of the solutions obtained and highlight the trade-off between such robustness and the price of robustness, providing useful insights for decision makers on the utilization of our newly developed models and algorithms.

The remainder of this paper is organized as follows. In Section 2, we introduce problem statements for the deterministic CTSNDP and the robust CTSNDP under travel time uncertainty. In Section 3, we present our new MILP model for the deterministic CTSNDP, and extend it to derive the robust optimization model and the robust satisficing model to incorporate travel time uncertainty. In Section 4, we illustrate our tailored C&CG algorithms for the robust optimization model and the robust satisficing model, respectively, and prove their correctness and convergence. The computational results are discussed in Section 5, followed by our concluding remarks in Section 6. All the proofs are presented in Appendix B.

2. Problem Statements

In this section, we first introduce the deterministic CTSNDP where travel times are given, and then define two variants of the robust CTSNDP where travel times are uncertain. The notation used for problem description and solution representation is summarized in Table A.1 in Appendix A.

2.1. Deterministic CTSNDP

The deterministic CTSNDP examined in this paper extends the problem setting in Boland et al. (2017), with shipment holding costs being incorporated. Unlike Boland et al. (2017), we define its feasible solutions over the physical network of the terminals, instead of the time-expanded network.

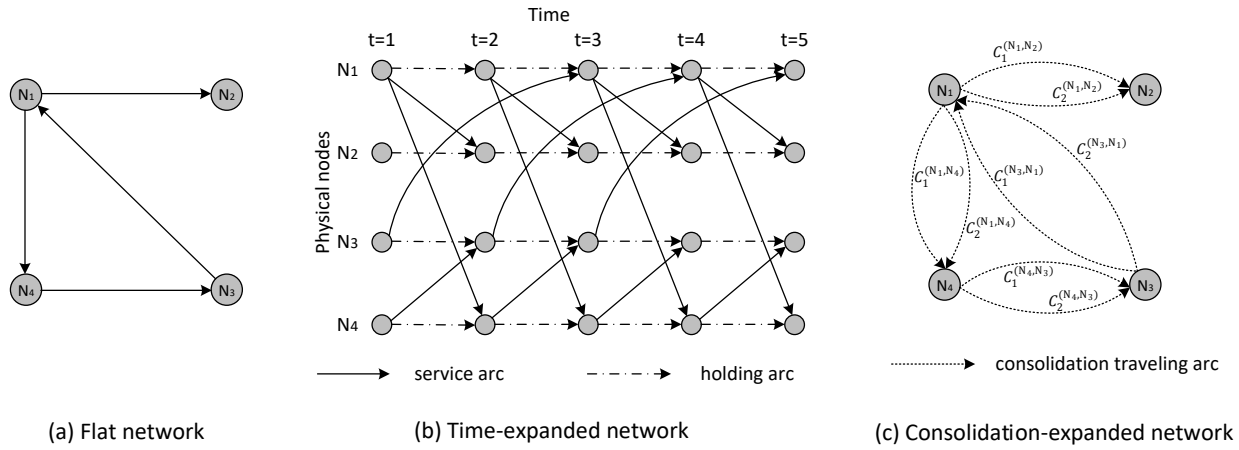
Consider a network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ with a physical node set $\mathcal{N}$ and a directed arc set $\mathcal{A}$, which is referred to as the *flat network*. Each physical node represents a terminal, and each arc represents a direct transport service from one terminal to another. Consider a commodity set $\mathcal{K}$, where each commodity $k \in \mathcal{K}$ represents a shipment, with its origin denoted by $o^k \in \mathcal{N}$, its destination denoted by $d^k \in \mathcal{N}$, and its shipping quantity denoted by $q^k \in \mathbb{N}_{>0}$. Each commodity $k \in \mathcal{K}$ has an earliest available time $e^k \in \mathbb{N}$ for departure from its origin o^k, and has a due time $l^k \in \mathbb{N}_{>0}$ for arrival at its destination d^k. No commodity can be delivered separately, and thus each commodity can only be assigned exactly one delivery path. As a result, each commodity must be picked up exactly once from the origin after the earliest available time and delivered exactly once to the destination before the due time. However, commodities can be temporarily stored at any nodes, waiting to be consolidated for shipping together on different arcs of the network.

In the flat network $\mathcal{D}$, each arc $(i, j) \in \mathcal{A}$ is associated with four attributes: (1) travel time $\tau_{ij} \in \mathbb{N}_{>0}$; (2) a per-unit-of-flow (travel) cost $c_{ij}^k \in \mathbb{R}_{>0}$ for each commodity $k \in \mathcal{K}$; (3) a fixed cost $f_{ij} \in \mathbb{R}_{>0}$ per vehicle for (shipping) service on the arc; and (4) a capacity $u_{ij} \in \mathbb{N}_{>0}$ per vehicle for (shipping) service on the arc. Additionally, both *in-transit* and *in-storage* holding costs are considered here for each commodity. In particular, the in-transit holding costs are incorporated into the flow costs c_{ij}^k for commodities $k \in \mathcal{K}$ and arcs $(i, j) \in \mathcal{A}$. A per-unit-of-demand-and-time in-storage holding cost $h^k \in \mathbb{R}_{\geq 0}$ is incurred when a commodity $k \in \mathcal{K}$ is stored at any node. It is worth noting that all the optimization models and solution methods derived in this paper can be extended to cases where the per-unit-of-demand-and-time in-storage holding costs depend not only on commodities $k \in \mathcal{K}$ but also on nodes $i \in \mathcal{N}$, by replacing h^k with h_i^k. These cases include the one where no holding cost is charged for each commodity at its destination.

The deterministic CTSNDP requires the deciding of delivery paths and consolidation plans for all commodities, as well as the numbers and the dispatch times of the required vehicles. Its objective is to satisfy all delivery requirements while keeping the total cost minimized.

A *feasible solution* to the deterministic CTSNDP consists of (i) a routing plan, (ii) a consolidation plan, and (iii) a departure schedule, these being defined as follows. We call a directed path P in the flat network $\mathcal{D}$ a *flat path*, which is represented by its node sequence $(\nu_1, \nu_2, \ldots, \nu_{m+1})$ and arc sequence $(a_1, a_2, \ldots, a_m)$, with $m \in \mathbb{N}_{>0}$ denoting the total number of its arcs. As in actual practice, the delivery path of each commodity cannot have repeated vertices or arcs, and thus must be an elementary flat path from the origin to the destination of the commodity. Accordingly, a *routing plan* $\mathcal{P}$ is defined as a collection of $|\mathcal{K}|$ elementary flat paths in the flat network $\mathcal{D}$, with each flat path $P^k \in \mathcal{P}$ for $k \in \mathcal{K}$ representing the delivery path of commodity k from its origin o^k to

Figure 1 **Examples of a flat network, a time-expanded network for 5 periods, and a consolidation-expanded network for $|\mathcal{K}| = 2$.**



(a) Flat network (b) Time-expanded network (c) Consolidation-expanded network

destination d^k, where the node and arc sequences of P^k are denoted by $(\nu_1^k, \nu_2^k, \ldots, \nu_{m^k+1}^k)$ and $(a_1^k, a_2^k, \ldots, a_{m^k}^k)$, respectively, with $\nu_1^k = o^k$ and $\nu_{m^k+1}^k = d^k$, and with no repeated nodes or arcs.

Given a routing plan $\mathcal{P}$ as defined above, we need to specify how commodities are consolidated on arcs of the flat network $\mathcal{D}$. For each $\alpha \in \mathcal{A}$, let $\mathcal{K}(\mathcal{P}, \alpha) = \{k \in \mathcal{K} \mid \exists a_n^k = \alpha, 1 \le n \le m^k\}$ indicate the subset of commodities whose flat paths in $\mathcal{P}$ pass through arc α. A consolidation on arc α for $\mathcal{P}$ can then be represented by a subset of $\mathcal{K}(\mathcal{P}, \alpha)$, so that commodities in this subset are consolidated to be shipped together on arc α, indicating how these commodities are transported on this arc for $\mathcal{P}$. Since each commodity cannot be delivered by separated paths of $\mathcal{P}$, there are at most $|\mathcal{K}|$ consolidations on each arc in any feasible solution of the deterministic CTSNDP. Accordingly, we can construct a *consolidation-expanded network* that shares the same nodes with the flat network and duplicates each arc in the flat network to create $|\mathcal{K}|$ consolidation traveling arcs. See Figure 1 for illustrative examples of the flat network, the time-expanded network, and the consolidation-expanded network.

For each arc $\alpha \in \mathcal{A}$ of the flat network $\mathcal{D}$, consider each consolidation traveling arc $\alpha^{(r)}$ of the consolidation-expanded network that is duplicated from α, where $r \in \{1, 2, \ldots, |\mathcal{K}|\}$. As demonstrated in Figure 1(c), we associate each $\alpha^{(r)}$ with $C_r^\alpha \subseteq \mathcal{K}(\mathcal{P}, \alpha)$ to indicate the r-th consolidation on arc α for the routing plan $\mathcal{P}$, so that all the commodities in C_r^α are shipped through arc α together. A *consolidation plan* $\mathcal{C}$ for $\mathcal{P}$ can thus be defined as a collection of consolidations $C_r^\alpha \subseteq \mathcal{K}(\mathcal{P}, \alpha)$ for $\alpha \in \mathcal{A}$ and $r \in \{1, 2, \ldots, |\mathcal{K}|\}$, where r is referred to as the *consolidation index*, and each consolidation C_r^α can be empty. If the consolidations C_r^α for $r = 1, 2, \ldots, |\mathcal{K}|$ cover all the commodities $k \in \mathcal{K}(\alpha)$ for each $\alpha \in \mathcal{A}$, i.e., $\bigcup_{r=1}^{|\mathcal{K}|} C_r^\alpha = \mathcal{K}(\alpha)$ is satisfied for each $\alpha \in \mathcal{A}$, then such a routing-consolidation pair $(\mathcal{P}, \mathcal{C})$ forms a *flat solution* to the deterministic CTSNDP.

Given a flat solution $(\mathcal{P},\mathcal{C})$, we need to further specify the departure time of each commodity from every node it passes through. Since each flat path in $\mathcal{P}$ is an elementary path, every commodity can depart from the same node at most once. Accordingly, a *departure schedule* $\mathcal{T}$ is defined as a collection of departure times $t^k_{\nu^k_n}$ for $k\in\mathcal{K}$ and $n\in\{1,2,\ldots,m^k\}$, indicating when commodity k departs from node ν^k_n of its flat path P^k. Thus, $(\mathcal{P},\mathcal{C},\mathcal{T})$ forms a *feasible solution* to the deterministic CTSNDP if the departure schedule $\mathcal{T}$ satisfies that

$$t^k_{\nu^k_n}\geq e^k,\quad \text{for } n=1, \tag{2.1}$$

$$t^k_{\nu^k_{n+1}}\geq t^k_{\nu^k_n}+\tau_{a^k_n},\ \text{for } n\in\{1,2,\ldots,m^k-1\}, \tag{2.2}$$

$$t^k_{\nu^k_n}+\tau_{a^k_n}\leq l^k,\ \text{for } n=m^k, \tag{2.3}$$

$$t^k_i=t^{k'}_i,\quad \text{for } k\in C^{(i,j)}_r \text{ and } k'\in C^{(i,j)}_r \text{ with } (i,j)\in\mathcal{A} \text{ and } r\in\{1,2,\cdots,|\mathcal{K}|\}. \tag{2.4}$$

Here, for each commodity $k\in\mathcal{K}$, constraints (2.1) and (2.3) together ensure that the departure time from its origin and arrival time at its destination are both within the time window $[e^k,l^k]$, and constraints (2.2) are due to the travel times of arcs on its flat path. Constraints (2.4) ensure that commodities consolidated on the same arc all pass the arc at the same time. A flat solution $(\mathcal{P},\mathcal{C})$ is *timely-implementable*, if there exists such a departure schedule $\mathcal{T}$ that satisfies (2.1)–(2.4).

From a feasible solution $(\mathcal{P},\mathcal{C},\mathcal{T})$, we can obtain holding times H^k_n for nodes ν^k_n with $n=1,2,\ldots,m^k+1$ on the flat path P^k of each commodity $k\in\mathcal{K}$:

$$H^k_n=\begin{cases} t^k_{\nu^k_n}-e^k, & \text{for } n=1,\\ t^k_{\nu^k_n}-(t^k_{\nu^k_{n-1}}+\tau_{a^k_{n-1}}), & \text{for } n\in\{2,\ldots,m^k\},\\ l^k-(t^k_{\nu^k_{n-1}}+\tau_{a^k_{n-1}}), & \text{for } n=m^k+1.\end{cases}$$

Accordingly, the total cost of solution $(\mathcal{P},\mathcal{C},\mathcal{T})$ can be represented as follows:

$$\sum_{\alpha\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}f_\alpha\lceil\frac{\sum_{k\in C^\alpha_r}q^k}{u_\alpha}\rceil+\sum_{k\in\mathcal{K}}\sum_{n=1}^{m^k}c^k_{a^k_n}q^k+\sum_{k\in\mathcal{K}}\sum_{n=1}^{m^k+1}h^kq^kH^k_n,$$

where the first term is the total fixed cost of vehicles needed, the second term is the total flow cost, and the third term is the total holding cost. It can be seen that the total fixed cost and flow cost depend only on the flat solution $(\mathcal{P},\mathcal{C})$, and the total holding cost depends only on the routing plan $\mathcal{P}$ and the departure schedule $\mathcal{T}$. Thus, we can define a function $f(\mathcal{P},\mathcal{C})$ to represent the total fixed cost and flow cost, and a function $h(\mathcal{P},\mathcal{T})$ to represent the total holding cost, where

$$f(\mathcal{P},\mathcal{C})=\sum_{\alpha\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}f_\alpha\lceil\frac{\sum_{k\in C^\alpha_r}q^k}{u_\alpha}\rceil+\sum_{k\in\mathcal{K}}\sum_{n=1}^{m^k}c^k_{a^k_n}q^k,$$

$$h(\mathcal{P},\mathcal{T})=\sum_{k\in\mathcal{K}}\sum_{n=1}^{m^k+1}h^kq^kH^k_n.$$

Without loss of generality, we assume that for each commodity $k \in \mathcal{K}$, the difference $(l^k - e^k)$ between its latest arrival time l^k at the destination and available time e^k at the origin is not smaller than the length of the shortest-time path from o^k to d^k in the flat network $\mathcal{D}$. This assumption is sufficient to ensure the existence of a feasible solution to the deterministic CTSNDP. The deterministic CTSNDP can thus be formulated as follows, where $\mathbb{D}$ indicates the domain of all the feasible solutions.

$$\text{(Deterministic CTSNDP)} \quad \min_{(\mathcal{P},\mathcal{C},\mathcal{T})\in\mathbb{D}} [f(\mathcal{P},\mathcal{C}) + h(\mathcal{P},\mathcal{T})].$$

2.2. Robust CTSNDP

We now introduce the robust CTSNDP under travel time uncertainty, which we refer to as the robust CTSNDP for short. Suppose that for each arc $\alpha \in \mathcal{A}$, the actual travel time $\tilde{\tau}_\alpha$ for commodities passing through α is determined by $\tilde{\tau}_\alpha = \overline{\tau}_\alpha + \hat{\tau}_\alpha \delta_\alpha$. Here, $\overline{\tau}_\alpha \in \mathbb{N}_{>0}$ is the *nominal* value of $\tilde{\tau}_\alpha$, and $\hat{\tau}_\alpha \in \mathbb{N}_0$ with $\hat{\tau}_\alpha < \overline{\tau}_\alpha$ is the maximum deviation of $\tilde{\tau}_\alpha$ with respect to the nominal value $\overline{\tau}_\alpha$. The coefficient δ_α is a random variable (but with unknown distribution), and its value falls within the range $[-1, 1]$. Thus, $\tilde{\tau}_\alpha$ falls within the range $[\overline{\tau}_\alpha - \hat{\tau}_\alpha, \overline{\tau}_\alpha + \hat{\tau}_\alpha]$.

For each $\alpha \in \mathcal{A}$, since there can be at most $|\mathcal{K}|$ consolidations on arc α, we use $\tilde{\tau}_{\alpha r}$ for $r \in \{1, 2, \ldots, |\mathcal{K}|\}$ to indicate the travel time of the r-th consolidation on arc α. Let $\mathbb{U}$ indicate the support of the vector $\boldsymbol{\delta}$ of random variables $\delta_{\alpha r}$ for $\alpha \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. We have that:

$$\mathbb{U} = \Big\{ \boldsymbol{\delta} : \delta_{\alpha r} \in [-1, 1], \forall \alpha \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\} \Big\}. \tag{2.5}$$

Moreover, for each realized coefficient value $\boldsymbol{\delta} \in \mathbb{U}$, we use $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ to indicate the vector of the corresponding realized travel times $(\overline{\tau}_\alpha + \hat{\tau}_\alpha \delta_{\alpha r})$ for $\alpha \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$, which can be defined as follows:

$$\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}) = \Big\{ \tilde{\boldsymbol{\tau}} : \tilde{\tau}_{\alpha r} = \overline{\tau}_\alpha + \hat{\tau}_\alpha \delta_{\alpha r}, \forall \alpha \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\} \Big\}. \tag{2.6}$$

In the remainder of this paper, we will use the term *scenario* to refer to vector $\boldsymbol{\delta}$, and we will refer to $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ as the realized travel time for scenario $\boldsymbol{\delta}$.

The decision process for the robust CTSNDP has two stages. In the first stage, which is before actual values of the travel times are realized, the problem needs to determine a routing plan $\mathcal{P}$ and a consolidation plan $\mathcal{C}$ that form a flat solution $(\mathcal{P}, \mathcal{C})$. Given $(\mathcal{P}, \mathcal{C})$, in the second stage, which is after the actual values of the travel times are realized, the problem needs to determine an actual departure schedule $\mathcal{T}$. Accordingly, $(\mathcal{P}, \mathcal{C})$ is a *here-and-now* decision which is independent of the realized travel times, and $\mathcal{T}$ is a *wait-and-see* decision which can adapt to the realized travel times.

Let us now consider the second stage of the robust CTSNDP. Given a flat solution $(\mathcal{P},\mathcal{C})$ determined in the first stage, and after actual travel times $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ with $\boldsymbol{\delta}\in\mathbb{U}$ are realized, one needs to determine an actual departure schedule $\mathcal{T}=(t^k_{\nu^k_n})_{k\in\mathcal{K},1\leq n\leq m^k}$, where each $t^k_{\nu^k_n}$ indicates the departure time of commodity k from node ν^k_n on the flat path P^k of $\mathcal{P}$. For each commodity $k\in\mathcal{K}$ and each arc $a^k_n=(\nu^k_n,\nu^k_{n+1})$ of P^k, since $(\mathcal{P},\mathcal{C})$ is a flat solution, there exists $r(k,n)\in\{1,2,\ldots,|\mathcal{K}|\}$ such that the commodity k is contained in the consolidation $C^{a^k_n}_{r(k,n)}$ of $\mathcal{C}$. This implies that the actual travel time of commodity k on arc a^k_n equals $\tilde{\tau}_{a^k_n,r(k,n)}$. Accordingly, the actual departure schedule $\mathcal{T}$ needs to satisfy constraints (2.1) due to the earliest available time e^k for $k\in\mathcal{K}$, constraints (2.4) due to the consolidations, and constraints (2.7) below

$$t^k_{\nu^k_{n+1}}\geq t^k_{\nu^k_n}+\tilde{\tau}_{a^k_n,r(k,n)},\ \text{for } k\in\mathcal{K},\ n\in\{1,2,\ldots,m^k-1\},\tag{2.7}$$

which are due to the actual travel times and are similar to the constraints (2.2) for $k\in\mathcal{K}$ with $\tau_{a^k_n}$ replaced by $\tilde{\tau}_{a^k_n,r(k,n)}$. As a result, the domain of such actual departure schedules $\mathcal{T}$ is denoted by $\mathbb{T}(\mathcal{P},\mathcal{C},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$.

Moreover, due to the travel time uncertainty, it will be costly to satisfy the due time constraints for every possible realization of the travel times. We therefore relax the due time constraints in the second stage of the robust CTSNDP. However, to restrict the violations of the due time constraints, we impose a penalty g^k per unit of time for the delay of each commodity k's arrival at its destination d^k. Let $g(\mathcal{P},\mathcal{T})$ indicate the total delay penalty for an actual departure schedule $\mathcal{T}$ with respect to flat paths in $\mathcal{P}$. We have that

$$g(\mathcal{P},\mathcal{T})=\sum_{k\in\mathcal{K}}g^k\cdot\max\{t^k_{\nu^k_{m^k}}+\tilde{\tau}_{a^k_{m^k},r(k,m^k)}-l^k,0\},$$

where $(t^k_{\nu^k_{m^k}}+\tilde{\tau}_{a^k_{m^k},r(k,m^k)})$ indicates the actual arrival time of commodity k at the destination d^k. Hence, under the realized travel times $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ with $\boldsymbol{\delta}\in\mathbb{U}$, the corresponding second-stage cost, including the holding costs and delay penalties, is determined by $\mathcal{P}$ and $\mathcal{T}$ and is equal to $h(\mathcal{P},\mathcal{T})+g(\mathcal{P},\mathcal{T})$. Its minimum value, $\min_{\mathcal{T}\in\mathbb{T}(\mathcal{P},\mathcal{C},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))}[h(\mathcal{P},\mathcal{T})+g(\mathcal{P},\mathcal{T})]\}$, is referred to as the second-stage cost of the robust CTSNDP under the scenario $\boldsymbol{\delta}$.

Remark 2.1 (Uncertainty Revelation) *Drawing from the literature on two-stage robust optimization problems (see, for example, Ben-Tal et al. 2004, Atamtürk and Zhang 2007, Yanıkoğlu et al. 2019), our statement above for the second stage of the robust CTSNDP assumes that all actual travel times are revealed before the departure schedule is determined. Nonetheless, in many practical scenarios, the actual travel time of certain commodities' transport through a particular*

arc cannot be revealed until the transport is completed. We refer to such an uncertainty revelation mechanism as a dynamic uncertainty revelation. Later we will show that solutions derived from our stated two-stage formulation of the robust CTSNDP can be adapted to situations under the dynamic uncertainty revelation, without increasing the total cost.

Next, let us consider the first stage of the robust CTSNDP. Before the actual travel times are realized, a flat solution $(\mathcal{P},\mathcal{C})$ needs to be determined. As commonly required in practice, such a flat solution $(\mathcal{P},\mathcal{C})$ needs to form a feasible solution to the deterministic CTSNDP under a nominal scenario, where travel times take their nominal values with no deviations. However, this cannot be ensured by the constraints imposed in the second stage of the robust CTSNDP, where the due time constraints are relaxed. As a result, we adopt a *light robustness* approach, originally proposed by Fischetti and Monaci (2009) for robust optimization, to establish the first stage of the robust CTSNDP. This approach requires that the first-stage decisions must be feasible to the deterministic CTSNDP under the nominal travel times. Accordingly, the flat solution $(\mathcal{P},\mathcal{C})$ to be determined in the first stage of the robust CTSNDP must be timely-implementable under the nominal scenario. In other words, there exists a departure schedule $\hat{\mathcal{T}}$ such that $(\mathcal{P},\mathcal{C},\hat{\mathcal{T}})$ forms a feasible solution to the deterministic CTSNDP under the nominal travel times. We refer to such a flat solution $(\mathcal{P},\mathcal{C})$ as a *nominal timely-implementable first-stage solution*, and use $\mathbb{F}$ to indicate the domain of all nominal timely-implementable first-stage solutions.

The robust CTSNDP aims to find a robust nominal timely-implementable first-stage solution under the travel time uncertainty. In this study, we adopt two modeling frameworks, namely robust optimization and robust sacrificing, to characterize the robustness of such solutions, which are illustrated in Sections 2.2.1 and 2.2.2, respectively. In Section 2.2.3, we will explain how solutions to our two-stage formulation of robust CTSNDP can be adapted to cases where uncertain travel times are revealed dynamically.

2.2.1. Robust Optimization Variant of CTSNDP Given an integer $\Gamma \in \mathbb{N}$, which is known as the *budget of uncertainty*, we can use it to adjust the level of robustness as needed. To achieve this, we define a budgeted uncertainty set $\mathbb{U}(\Gamma)$ as follows on the random scenario $\boldsymbol{\delta}$ that determines travel times:

$$\mathbb{U}(\Gamma) = \Big\{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_1 \leq \Gamma, \delta_{\alpha r} \in [-1,1], \forall \alpha \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\}\Big\}. \tag{2.8}$$

It contains all the possible realizations of $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_1$, which equals $\sum_{\alpha\in\mathcal{A},r\in\{1,2,...,|\mathcal{K}|\}} |\delta_{\alpha r}|$ and represents the total relative deviation of the travel times $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ from their nominal values with respect to their maximum deviations, does not exceed the given budget of uncertainty Γ.

The robust optimization variant of the CTSNDP under travel time uncertainty (or RO-CTSNDP in short) has an objective to minimize the worst-case total two-stage cost with respect to the budgeted uncertainty set $\mathbb{U}(\Gamma)$ on $\boldsymbol{\delta}$. To achieve this, the RO-CTSNDP needs to determine a nominal timely-implementable first-stage solution $(\mathcal{P},\mathcal{C}) \in \mathbb{F}$ that minimizes the sum of the first-stage cost (which is independent of the realization of $\boldsymbol{\delta}$) and the worst-case second-stage cost (which is over the budgeted uncertainty set $\mathbb{U}(\Gamma)$ on $\boldsymbol{\delta}$). Accordingly, the RO-CTSNDP can be formulated as follows:

$$[\text{RO-CTSNDP}] \quad \min_{(\mathcal{P},\mathcal{C})\in\mathbb{F}} \{f(\mathcal{P},\mathcal{C}) + \max_{\boldsymbol{\delta}\in\mathbb{U}(\Gamma)} \min_{\mathcal{T}\in\mathbb{T}(\mathcal{P},\mathcal{C},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))} [h(\mathcal{P},\mathcal{T}) + g(\mathcal{P},\mathcal{T})]\}.$$

2.2.2. Robust Satisficing Variant of CTSNDP We follow the modeling framework proposed by Long et al. (2023) to establish the robust satisficing variant of the CTSNDP under travel time uncertainty (or RS-CTSNDP in short). Let $\mathcal{Z}_0$ represent the optimal objective value of the deterministic CTSNDP under nominal travel times. Given a prescribed target $\mathcal{Z}$ of the total two-stage cost with $\mathcal{Z} \geq \mathcal{Z}_0$, the RS-CTSNDP needs to determine a nominal timely-implementable first-stage solution $(\mathcal{P},\mathcal{C}) \in \mathbb{F}$ that best achieves the prescribed target $\mathcal{Z}$, so that the worst-case normalized magnitude of the deviation from the target of the total two-stage cost is minimized. Accordingly, the RS-CTSNDP can be formulated as follows:

$$\min_{(\mathcal{P},\mathcal{C})\in\mathbb{F}} \{\rho \in \mathbb{R}_{\geq 0} : f(\mathcal{P},\mathcal{C}) + \min_{\mathcal{T}\in\mathbb{T}(\mathcal{P},\mathcal{C},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))} [h(\mathcal{P},\mathcal{T}) + g(\mathcal{P},\mathcal{T})] - \mathcal{Z} \leq \rho\|\boldsymbol{\delta}\|_1,\ \forall \boldsymbol{\delta} \in \mathbb{U}\}.$$

Here, the constraints imposed on the first-stage solution $(\mathcal{P},\mathcal{C}) \in \mathbb{F}$ restrict the deviation of the total two-stage cost from the prescribed target $\mathcal{Z}$ to not exceed $\rho\|\boldsymbol{\delta}\|_1$ for every possible scenario $\boldsymbol{\delta}$ in the uncertainty set $\mathbb{U}$. As a result, ρ indicates the worst-case magnitude of the deviation from the prescribed cost target, normalized by the total relative deviation $\|\boldsymbol{\delta}\|_1$ of the travel times. This quantity measures the fragility of a given solution and needs to be minimized to attain robustness.

2.2.3. Adaption to Dynamic Uncertainty Revelation of Travel Times We will now show that solutions to our two-stage formulation of the robust CTSNDP, where the second stage assumes that all actual travel times are revealed before the actual departure schedule is determined, can be adapted to cases under the dynamic uncertainty revelation.

Consider any first-stage solution $(\mathcal{P},\mathcal{C})$ of a routing plan and a consolidation plan. Under dynamic uncertainty revelation, the actual travel time of each consolidated shipment on an arc $(i,j) \in \mathcal{A}$ is only revealed after its arrival at node j. One possible approach to determine the departure schedule is to apply a *reactive policy* in which for each $(i,j) \in \mathcal{A}$, every consolidated shipment on arc (i,j) departs from node i as soon as all the commodities for the shipment have arrived at node i. Consequently, the departure times of consolidated shipments depend only on the actual travel

times that have been realized in the previous part of their transport, and not on the future travel times that are yet to be revealed.

Next, consider any possible scenario $\boldsymbol{\delta}$, which determines travel times $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$. We are going to show that the departure schedule obtained by the reactive policy introduced above achieves the minimum second stage cost under $\boldsymbol{\delta}$. On the one hand, the reactive policy ensures that for each arc $(i,j) \in \mathcal{A}$, every consolidated shipment on arc (i,j) departs from node i as soon as all the commodities for the shipment have arrived at node i, thus guaranteeing that all commodities arrive at their destinations at the earliest possible time. Therefore, the total delay penalty must be minimized. On the other hand, for each commodity k, let $\tilde{T}_k(\mathcal{P},\mathcal{C},\boldsymbol{\delta})$ indicate the total travel time of commodity k under $(\mathcal{P},\mathcal{C})$ and $\boldsymbol{\delta}$. Under the reactive policy, it can be seen that the total in-storage holding time equals $\max\{l_k - e_k - \tilde{T}_k(\mathcal{P},\mathcal{C},\boldsymbol{\delta}), 0\}$, which achieves the minimum total in-storage holding time. Thus, the reactive policy achieves a minimum total in-storage holding cost that equals $h^k \max\{l_k - e_k - \tilde{T}_k(\mathcal{P},\mathcal{C},\boldsymbol{\delta}), 0\}$.

Therefore, the departure schedule derived from the reactive policy achieves the minimum total second-stage cost for each possible $\boldsymbol{\delta}$. Based on this, it is evident that solutions to the RO-CTSNDP and the RS-CTSNDP can be adapted by the reactive policy to cases under the dynamic uncertainty revelation without increasing their objective values.

Remark 2.2 (Node-Dependent Holding Costs) *It is worth noting that when the per-unit-of-demand-and-time in-storage holding costs depend not only on commodities $k \in \mathcal{K}$ but also on nodes $i \in \mathcal{N}$, the notation h^k in $h(\mathcal{P},\mathcal{T})$ needs to be replaced with h_i^k. As a result, although optimization models and solution methods derived in this paper can be extended, the reactive policy illustrated above may not guarantee to achieve the minimum total second-stage cost for each possible $\boldsymbol{\delta}$. Nevertheless, by following the same argument above, the reactive policy still ensures attainment of the minimum total second-stage cost for every possible $\boldsymbol{\delta}$, when the holding costs h_n^k for each commodity $k \in \mathcal{K}$ are equal to h^k for all nodes n that are not the destination d^k of commodity k, and h_n^k is less than or equal to h^k if $n = d^k$.*

3. Optimization Models

In this section, we first propose a novel compact MILP model for the deterministic CTSNDP. It is based on the consolidation-expanded network rather than the time-expanded network, and utilizes consolidation indices rather than time indices. Leveraging this new model, we proceed to develop two-stage mixed-integer nonlinear programming models (MINLP) for the two variants of the robust CTSNDP, namely the RO-CTSNDP and the RS-CTSNDP. For both models, their second stage

cost can be computed by solving a linear program (LP), enabling us to develop efficient solution methods that will be presented later in Section 4. The notation used for various optimization models is summarized in Table A.2 in Appendix A.

3.1. Consolidation-Indexed MILP Model for Deterministic CTSNDP

Our new MILP model for the deterministic CTSNDP utilizes consolidation indices instead of time indices to define decision variables and formulate constraints related to consolidations.

According to the problem description in Section 2, a feasible solution to the deterministic CTSNDP consists of a routing plan $\mathcal{P}$, a consolidation plan $\mathcal{C}$, and a departure schedule $\mathcal{T}$. To represent the routing plan $\mathcal{P}$, we introduce a binary variable x_{ij}^k for each $(i,j) \in \mathcal{A}$ and $k \in \mathcal{K}$, indicating whether commodity $k \in \mathcal{K}$ passes through arc (i,j).

To represent the consolidation plan $\mathcal{C}$, we first introduce a binary variable z_{ijr}^k for each $(i,j) \in \mathcal{A}$, $r \in \{1,2,\cdots,|\mathcal{K}|\}$, and $k \in \mathcal{K}$, indicating whether the r-th consolidation $C_r^{(i,j)}$ on arc (i,j) contains commodity k. We then introduce a non-negative integer variable y_{ijr} for each $(i,j) \in \mathcal{A}$ and $r \in \{1,2,\cdots,|\mathcal{K}|\}$, indicating the number of vehicles needed by consolidation $C_r^{(i,j)}$ of arc (i,j) to accommodate the commodities in consolidation $C_r^{(i,j)}$.

To represent the departure schedule $\mathcal{T}$, we first introduce a non-negative continuous variable v_{ij}^k for each $(i,j) \in \mathcal{A}$ and $k \in \mathcal{K}$, which indicates the time when commodity k departs from node i when passing through arc (i,j). If commodity k does not pass through arc (i,j), then v_{ij}^k equals 0. We then introduce a non-negative continuous variable b_{ijr} for each $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$, which represents the time when commodities of the r-th consolidation $C_r^{(i,j)}$ on arc (i,j) depart from node i. We also introduce a non-negative continuous variable w_i^k for $i \in \mathcal{N}$ and $k \in \mathcal{K}$ to represent the holding time for commodity k at terminal i. It equals 0 if commodity k does not pass node i.

Accordingly, the deterministic CTSNDP can be represented by the following compact MILP model, referred to as model DO, where M denotes a sufficiently large constant:

$$\text{[DO]} \quad \min \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h^k q^k) \cdot w_i^k \tag{3.1}$$

$$\text{s.t.} \sum_{(i,j)\in\mathcal{A}} x_{ij}^k - \sum_{(j,i)\in\mathcal{A}} x_{ji}^k = \begin{cases} 1, & i = o^k, \\ -1, & i = d^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall k \in \mathcal{K}, i \in \mathcal{N}, \tag{3.2}$$

$$\sum_{k\in\mathcal{K}} q^k z_{ijr}^k \le u_{ij} y_{ijr}, \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\}, \tag{3.3}$$

$$\sum_{k\in\mathcal{K}} q^k z_{ijr}^k \ge u_{ij} y_{ijr} - u_{ij} + 1, \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\}, \tag{3.4}$$

$$\sum_{r=1}^{|\mathcal{K}|} z_{ijr}^k = x_{ij}^k, \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, \tag{3.5}$$

$$\sum_{j:(j,i)\in\mathcal{A}}(v^k_{ji}+\tau_{ji}x^k_{ji})\le\sum_{j:(i,j)\in\mathcal{A}}v^k_{ij},\qquad\forall\ i\in\mathcal{N}\setminus\{o^k,d^k\},k\in\mathcal{K},\tag{3.6}$$

$$\sum_{j:(o^k,j)\in\mathcal{A}}v^k_{o^kj}\ge e^k,\qquad\forall\ k\in\mathcal{K},\tag{3.7}$$

$$\sum_{j:(j,d^k)\in\mathcal{A}}(v^k_{jd^k}+\tau_{jd^k}x^k_{jd^k})\le l^k,\qquad\forall\ k\in\mathcal{K},\tag{3.8}$$

$$v^k_{ij}\le Mx^k_{ij},\qquad\forall\ (i,j)\in\mathcal{A},k\in\mathcal{K},\tag{3.9}$$

$$v^k_{ij}\le b_{ijr}+M(1-z^k_{ijr}),\qquad\forall\ (i,j)\in\mathcal{A},k\in\mathcal{K},r\in\{1,2,...,|\mathcal{K}|\},\tag{3.10}$$

$$v^k_{ij}\ge b_{ijr}-M(1-z^k_{ijr}),\qquad\forall\ (i,j)\in\mathcal{A},k\in\mathcal{K},r\in\{1,2,...,|\mathcal{K}|\},\tag{3.11}$$

$$w^k_i=\begin{cases}\sum\limits_{j:(i,j)\in\mathcal{A}}v^k_{ij}-e^k, & i=o^k,\\ l^k-\sum\limits_{j:(j,i)\in\mathcal{A}}(v^k_{ji}+\tau_{ji}x^k_{ji}), & i=d^k,\\ \sum\limits_{j:(i,j)\in\mathcal{A}}v^k_{ij}-\sum\limits_{j:(j,i)\in\mathcal{A}}(v^k_{ji}+\tau_{ji}x^k_{ji}), & \text{otherwise,}\end{cases}\qquad\forall\ i\in\mathcal{N},\forall\ k\in\mathcal{K},\tag{3.12}$$

$$x^k_{ij}\in\{0,1\},\qquad\forall\ (i,j)\in\mathcal{A},k\in\mathcal{K},\tag{3.13}$$

$$y_{ijr}\in\mathbb{N}_{\ge0},\qquad\forall\ (i,j)\in\mathcal{A},r\in\{1,2,...,|\mathcal{K}|\},\tag{3.14}$$

$$z^k_{ijr}\in\{0,1\},\qquad\forall\ (i,j)\in\mathcal{A},k\in\mathcal{K},r\in\{1,2,...,|\mathcal{K}|\},\tag{3.15}$$

$$v^k_{ij}\ge0,\qquad\forall\ (i,j)\in\mathcal{A},k\in\mathcal{K},\tag{3.16}$$

$$b_{ijr}\ge0,\qquad\forall\ (i,j)\in\mathcal{A},r\in\{1,2,...,|\mathcal{K}|\},\tag{3.17}$$

$$w^k_i\ge0,\qquad\forall\ i\in\mathcal{N},k\in\mathcal{K}.\tag{3.18}$$

In model DO, the objective function (3.1) indicates the total cost to be minimized, which includes three terms for the total fixed cost, total flow cost, and total holding cost, respectively. Constraints (3.2)–(3.5) are imposed to define the routing and the consolidation plans. Specifically, constraints (3.2) are *flow balance constraints*, ensuring that each commodity travels along one flat path from its origin to its destination. Constraints (3.3) and (3.4) are *capacity constraints*. They ensure that the total quantity of commodities in each consolidation of an arc does not exceed the total capacity of the vehicles assigned to each consolidation of the arc, and restrict that $y_{ijr}=\lceil(\sum_{k\in\mathcal{K}}q^kz^k_{ijr})/u_{ij}\rceil$ for every $(i,j)\in\mathcal{A}$, $r\in\{1,2,...,|\mathcal{K}|\}$, which equals the number of vehicles needed by consolidation $C^{(i,j)}_r$ of arc (i,j). Constraints (3.5) are *consolidation coverage constraints*, ensuring that for every arc (i,j) on the flat path of commodity $k\in\mathcal{K}$, there exists a consolidation of arc (i,j) that contains k. Constraints (3.6)–(3.11) are imposed to define the departure schedule. Specifically, constraints (3.6)–(3.8) are imposed on commodities' departure times with respect to the travel time of each arc, as well as the earliest available time and the due time of each commodity. Constraints (3.9) ensure that for each commodity, its departure time from each of its unvisited nodes is zero. Constraints

(3.10) and (3.11) ensure that for each arc $(i,j) \in \mathcal{A}$, the commodities that are consolidated to be shipped together through (i,j) have the same departure time from node i. Constraints (3.12) are imposed to define the holding time for each commodity $k \in \mathcal{K}$ and each node $i \in \mathcal{N}$, based on the departure schedule and the routing plan. The variables w_i^k and constraints (3.12) clearly show that the model DO can be extended to incorporate node-dependent holding costs. Constraints (3.13)-(3.18) define the domains of all the decision variables.

For each feasible solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w})$ of model DO, $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ corresponds to a flat solution $(\mathcal{P}, \mathcal{C})$, and $\boldsymbol{v}$ corresponds to a departure schedule $\mathcal{T}$ that satisfies (2.1)–(2.4), which imply that such $(\mathcal{P}, \mathcal{C}, \mathcal{T})$ forms a feasible solution to the deterministic CTSNDP. As far as we know, model DO is the first compact MILP model of the deterministic CTSNDP that utilizes consolidation indices, and thus, we refer to it as the *consolidation-indexed* MILP model of the deterministic CTSNDP.

3.2. Two-Stage MINLP Optimization Models for Robust CTSNDP

Our newly proposed model DO of the deterministic CTSNDP eliminates the need for time indices. As a result, it can be extended to derive two-stage MINLP models for two variants of the robust CTSNDP, namely the RO-CTSNDP and the RS-CTSNDP, so that the second stage cost can be computed by solving a linear program.

According to the problem statements in Section 2, for both the two variants of the robust CTSNDP, $(\boldsymbol{x}, \boldsymbol{z})$ of the first-stage decisions needs to ensure the existence of a departure schedule that satisfies the constraints with respect to commodities' earliest available times and due times under the nominal scenario. For this, we need to introduce decision variables $\overline{v}_{ij}^k$ and $\overline{b}_{ijr}$ to indicate commodities' departure times and consolidations' departure times for the nominal scenario, similar to the variables v_{ij}^k and b_{ijr} of model DO. Moreover, in the second stage, constraints with respect to the commodities' due times are relaxed, but delay penalties are imposed. As a result, we need to introduce an additional decision variable s^k for each $k \in \mathcal{K}$, indicating the delay of commodity k's arrival at its destination.

3.2.1. Robust Optimization Model According to the problem statement in Section 2.2.1, the RO-CTSNDP can be formulated as the following two-stage MINLP, referred to as model RO, where M denotes a sufficiently large constant:

$$\text{[RO]} \ \min \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + F_{RP}(\boldsymbol{x}, \boldsymbol{z}) \tag{3.19}$$

$$\text{s.t. } (3.2) - (3.5), (3.13) - (3.15) \tag{3.20}$$

$$\sum_{j:(j,i)\in\mathcal{A}} (\overline{v}_{ji}^k + \overline{\tau}_{ji} x_{ji}^k) \leq \sum_{j:(i,j)\in\mathcal{A}} \overline{v}_{ij}^k, \qquad \forall\, i \in \mathcal{N} \setminus \{o^k, d^k\}, k \in \mathcal{K}, \tag{3.21}$$

$$\sum_{j:(o^k,j)\in\mathcal{A}} \overline{v}^k_{o^k j} \geq e^k, \qquad \forall\ k\in\mathcal{K}, \tag{3.22}$$

$$\sum_{j:(j,d^k)\in\mathcal{A}} (\overline{v}^k_{jd^k} + \overline{\tau}_{jd^k} x^k_{id^k}) \leq l^k, \qquad \forall\ k\in\mathcal{K}, \tag{3.23}$$

$$\overline{v}^k_{ij} \leq M x^k_{ij}, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.24}$$

$$\overline{v}^k_{ij} \leq \overline{b}_{ijr} + M(1-z^k_{ijr}), \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.25}$$

$$\overline{v}^k_{ij} \geq \overline{b}_{ijr} - M(1-z^k_{ijr}), \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.26}$$

$$\overline{v}^k_{ij} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.27}$$

$$\overline{b}_{ijr} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}. \tag{3.28}$$

Here, $F_{RP}(\boldsymbol{x},\boldsymbol{z})$ indicates the worst-case second-stage cost and can be calculated by the following max-min optimization model, which is referred to as model RP$(\boldsymbol{x},\boldsymbol{z})$, where M_1 denotes a sufficiently large constant.

$$[\mathrm{RP}(\boldsymbol{x},\boldsymbol{z})] \quad F_{RP}(\boldsymbol{x},\boldsymbol{z}) = \max_{\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}):\boldsymbol{\delta}\in\mathbb{U}(\Gamma)} \ \min \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}} (h^k q^k)\cdot w^k_i + \sum_{k\in\mathcal{K}} g^k\cdot s^k \tag{3.29}$$

$$\text{s.t.} \sum_{j:(j,i)\in\mathcal{A}} (v^k_{ji} + \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z^k_{jir}) \leq \sum_{j:(i,j)\in\mathcal{A}} v^k_{ij}, \qquad \forall\ i\in\mathcal{N}\setminus\{o^k,d^k\}, k\in\mathcal{K}, \tag{3.30}$$

$$\sum_{j:(o^k,j)\in\mathcal{A}} v^k_{o^k j} \geq e^k, \qquad \forall\ k\in\mathcal{K}, \tag{3.31}$$

$$\sum_{j:(j,d^k)\in\mathcal{A}} (v^k_{jd^k} + \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jd^k r} z^k_{jd^k r}) \leq l^k + s^k, \qquad \forall\ k\in\mathcal{K}, \tag{3.32}$$

$$v^k_{ij} \leq M_1 x^k_{ij}, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.33}$$

$$v^k_{ij} \leq b_{ijr} + M_1(1-z^k_{ijr}), \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.34}$$

$$v^k_{ij} \geq b_{ijr} - M_1(1-z^k_{ijr}), \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.35}$$

$$w^k_i \geq \begin{cases} \sum\limits_{j:(i,j)\in\mathcal{A}} v^k_{ij} - e^k, & i = o^k, \\ (l^k + s^k) - \sum\limits_{j:(j,i)\in\mathcal{A}} (v^k_{ji} + \sum\limits_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z^k_{jir}), & i = d^k, \\ \sum\limits_{j:(i,j)\in\mathcal{A}} v^k_{ij} - \sum\limits_{j:(j,i)\in\mathcal{A}} (v^k_{ji} + \sum\limits_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z^k_{jir}), & \text{otherwise}, \end{cases} \quad \forall\ i\in\mathcal{N}, \forall\ k\in\mathcal{K}, \tag{3.36}$$

$$v^k_{ij} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.37}$$

$$b_{ijr} \geq 0, \qquad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.38}$$

$$w^k_i \geq 0, \qquad \forall i\in\mathcal{N}, k\in\mathcal{K}, \tag{3.39}$$

$$s^k \geq 0, \qquad \forall k\in\mathcal{K}. \tag{3.40}$$

The objective (3.19) of model RO is to minimize the sum of the deterministic first-stage cost and the worst-case second-stage cost with respect to the uncertainty set $\mathbb{U}(\Gamma)$ on $\boldsymbol{\delta}$. The first-stage cost includes the fixed costs and the flow costs shown in the first two terms of (3.19). The worst-case second-stage cost is represented by $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$. In model RO, constraints in (3.20) are the same as those of model DO imposed on $(\boldsymbol{x}, \boldsymbol{z})$. Constraints (3.21)–(3.28) are similar to (3.6)–(3.11), (3.16), and (3.17) of model DO, with τ_{ji} replaced by the nominal travel times $\overline{\tau}_{ji}$. These constraints are imposed to ensure the existence of a feasible departure schedule under the nominal scenario.

The max-min optimization model RP$(\boldsymbol{x}, \boldsymbol{z})$ is to compute the worst-case second-stage cost for $(\boldsymbol{x}, \boldsymbol{z})$. Given any $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ with $\boldsymbol{\delta} \in \mathbb{U}(\Gamma)$, the inner minimization problem of RP$(\boldsymbol{x}, \boldsymbol{z})$ needs to determine $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{s})$, with the objective of minimizing the second-stage cost that equals the sum of the holding costs and delay penalties as shown in (3.29). Most of the constraints in the inner minimization problem are the same as those of model DO imposed on $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w})$, except (3.30), (3.32) and (3.36). Compared with constraints (3.6), (3.8), and (3.12) of model DO, constraints (3.30), (3.32), and (3.36) replace $\tau_{ji} x^k_{ji}$ with $\sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z^k_{jir}$ for each $(j,i) \in \mathcal{A}$, as the latter indicates the actual travel time of commodity k on arc (j,i) if k passes through (j,i). Moreover, the decision variable s^k for $k \in \mathcal{K}$ is included in the right-hand sides of constraints (3.32) and (3.36), in order to represent the delay in commodity k's arrival at its destination.

Let $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ denote the optimal objective value of the inner minimization problem of model RP$(\boldsymbol{x}, \boldsymbol{z})$, which is the second-stage cost. This can be calculated by the following linear program, which is referred to as model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$:

$$[\mathrm{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))] \quad F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) = \min \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} (h^k q^k) \cdot w^k_i + \sum_{k \in \mathcal{K}} g^k \cdot s^k \tag{3.41}$$

$$\text{s.t.} \quad (3.30) - (3.40). \tag{3.42}$$

3.2.2. Robust Satisficing Model According to the problem statement in Section 2.2.2, the RS-CTSNDP can also be formulated as a two-stage MINLP, which is shown below and is referred to as model RS:

$$[\mathrm{RS}] \ \min \ \rho \tag{3.43}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c^k_{ij} q^k) \cdot x^k_{ij} + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} \le \rho \|\boldsymbol{\delta}\|_1, \ \forall \ \boldsymbol{\delta} \in \mathbb{U}, \tag{3.44}$$

$$\rho \ge 0, \tag{3.45}$$

$$(3.2) - (3.5), (3.13) - (3.15), (3.21) - (3.28). \tag{3.46}$$

Model RS aims to minimize ρ, which represents the worst-case magnitude of the deviation from the prescribed cost target, normalized by the total relative deviation $\|\boldsymbol{\delta}\|_1$ of the travel times. Here, $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ is the optimal objective value of model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ defined in (3.41) and (3.42), indicating the second stage cost of $(\boldsymbol{x}, \boldsymbol{z})$ under any given $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ with $\boldsymbol{\delta}$ in the uncertainty set $\mathbb{U}$. In model RS, constraints (3.44) specify that for every possible scenario $\boldsymbol{\delta}$, the deviation of the total two-stage cost from the prescribed target $\mathcal{Z}$ cannot exceed $\rho\|\boldsymbol{\delta}\|_1$. Constraint (3.45) defines the domain of variable ρ. Other constraints in (3.46) are the same as those in model RO, as they are imposed to ensure that $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ forms a nominal timely-implementable first-stage solution.

4. Exact Algorithms

In this section, we develop two solution algorithms to solve model RO and model RS, respectively. They both follow a column-and-constraint generation (C&CG) framework proposed by Zeng and Zhao (2013), which has been successfully utilized in solving two-stage robust optimization models for diverse problem domains (see, for example, Zeng and Zhao 2013, Wang and Qi 2020).

For both model RO and model RS, we first reformulate them into a noncompact MILP, incorporating variables and constraints for every possible scenario $\boldsymbol{\delta}$ within their corresponding uncertainty sets. This allows us to obtain a relaxation MILP for any subset of their uncertainty sets. Our C&CG algorithms then solve such a relaxation MILP (referred to as a master problem) in each iteration, with respect to a current subset of the uncertainty set. This provides both a lower bound on the optimal objective value and a first-stage solution. From the first-stage solution, we can further derive an upper bound on the optimal objective value. If the upper and lower bounds are equal, the first-stage solution implies an optimal solution, and our C&CG algorithms terminate. If not, we need to identify a new possible scenario $\boldsymbol{\delta}$ and add it to the current subset of the uncertainty set, so that the relaxation MILP is extended and strengthened with new decision variables and constraints. Our C&CG algorithms then proceed to the next iteration.

The critical step of our C&CG algorithm is the solution to the subproblem. For model RO, as in many existing studies, its subproblem can be formulated as an optimization model with a bi-linear objective function. Accordingly, we can utilize an integral property of the budgeted uncertainty set to reformulate the subproblem as an MILP model, and can then solve it directly using an optimization solver.

For model RS, its subproblem requires fractional optimization and cannot be formulated either as an optimization model with a bi-linear objective function or as an MILP model. To overcome this challenge, we have developed an enhanced bisection search procedure. In each iteration of the procedure, an MILP model is established and solved by an optimization solver. With this, we

demonstrate for the first time in the literature that the C&CG solution framework can be applied to solving a two-stage robust satisficing problem, which has promising potential for also solving other similar problems.

In the remainder of this section, we are going to illustrate our C&CG algorithms for model RO and model RS, respectively, and show their correctness and convergence. Additional information regarding various acceleration strategies employed in our algorithm implementation is provided in Appendix C.

4.1. C&CG Algorithm for Robust Optimization Model

Let $\mathcal{X}$ denote the domain of variables $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}})$ defined by linear constraints (3.20)–(3.28). Let $\mathcal{Q}(\boldsymbol{\delta})$ denote the domain of variables $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{s})$ defined by linear constraints (3.30)–(3.40) under the realized travel time $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ for scenario $\boldsymbol{\delta} \in \mathbb{U}(\Gamma)$. Accordingly, model RO proposed in Section 3.2.1 can be rewritten as the following noncompact MILP, referred to as model ROMILP:

$$\text{[ROMILP]} \ \min \quad \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + \phi \tag{4.1}$$

$$s.t. \ \ \phi \geq \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h^k q^k) \cdot w_i^{k(\boldsymbol{\delta})} + \sum_{k\in\mathcal{K}} g^k \cdot s^{k(\boldsymbol{\delta})}, \quad \forall\ \boldsymbol{\delta} \in \mathbb{U}(\Gamma), \tag{4.2}$$

$$(\boldsymbol{v}^{(\boldsymbol{\delta})}, \boldsymbol{b}^{(\boldsymbol{\delta})}, \boldsymbol{w}^{(\boldsymbol{\delta})}, \boldsymbol{s}^{(\boldsymbol{\delta})}) \in \mathcal{Q}(\boldsymbol{\delta}), \quad \forall\ \boldsymbol{\delta} \in \mathbb{U}(\Gamma), \tag{4.3}$$

$$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}}) \in \mathcal{X}. \tag{4.4}$$

Here, ϕ is a newly introduced decision variable, and $(\boldsymbol{v}^{(\boldsymbol{\delta})}, \boldsymbol{b}^{(\boldsymbol{\delta})}, \boldsymbol{w}^{(\boldsymbol{\delta})}, \boldsymbol{s}^{(\boldsymbol{\delta})})$ represents a vector of second-stage decision variables associated with each possible scenario $\boldsymbol{\delta}$ in $\mathbb{U}(\Gamma)$. Constraints (4.2) and (4.3) ensure that ϕ equals the worst-case second-stage cost. As a result, solving the min-max-min model RO is reduced to solving the above noncompact MILP model ROMILP.

Model ROMILP can be relaxed by replacing $\mathbb{U}(\Gamma)$ in constraints (4.2) and (4.3) with any of its subsets $\Lambda \subseteq \mathbb{U}(\Gamma)$. The resulting relaxation is referred to as model ROMILP(Λ). The relaxation can be strengthened by appending to Λ more possible scenarios $\boldsymbol{\delta}$ in $\mathbb{U}(\Gamma)$. When Λ equals $\mathbb{U}(\Gamma)$, model ROMILP(Λ) and model ROMILP are equivalent.

Accordingly, our C&CG algorithm for model RO, which is referred to as the RO-C&CG algorithm, iteratively solves model ROMILP(Λ) to obtain a first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and appends its worst-case scenario $\boldsymbol{\delta}$ to Λ, until $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ implies an optimal solution. With respect to model RO, a scenario $\boldsymbol{\delta}$ is a worst-case scenario, if under this value, the second-stage cost of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ equals the worst-case second-stage cost $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$.

We next illustrate our computation of the worst-case scenario $\boldsymbol{\delta}$ for any first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and then provide details of our RO-C&CG algorithm for model RO.

4.1.1. Computing the worst-case scenario $\boldsymbol{\delta}$ For any first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, both its worst-case second-stage cost $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ and the corresponding worst-case scenario $\boldsymbol{\delta}$ can be determined by solving the max-min model defined by (3.29)–(3.40), which, however, is difficult to solve directly. To overcome this difficulty, we need to reformulate the max-min model to an equivalent maximization MILP model as illustrated below, which is much more tractable.

First, consider the inner minimization problem of the max-min model defined by (3.29)–(3.40) under any given $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ with $\boldsymbol{\delta} \in \mathbb{U}(\Gamma)$. It forms a linear program, as shown in model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ in (3.41)–(3.42). Lemma 4.1 below indicates that model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ always has a feasible solution for each $(\boldsymbol{x}, \boldsymbol{z})$ that satisfies constraints (3.20)–(3.28) of model RO and for each $\boldsymbol{\delta} \in \mathbb{U}(\Gamma)$, which implies that the recourse of model RO is relatively complete.

Lemma 4.1 *For any $(\boldsymbol{x}, \boldsymbol{z})$ that satisfies constraints (3.20)–(3.28) of model RO, and for any $\boldsymbol{\delta} \in \mathbb{U}(\Gamma)$, model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ in (3.41)–(3.42) always has a feasible solution.*

Next, for model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$, let β_i^k, γ^k, ψ^k, η_{ij}^k, θ_{ijr}^k, ξ_{ijr}^k, and λ_i^k denote the dual variables associated with its constraints (3.30)–(3.36), respectively. By Lemma 4.1 and the strong duality theorem, the optimal objective value of LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$ equals that of its dual linear program below, which we refer to as model DLP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$:

$$
\begin{aligned}
[\mathrm{DLP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})] \quad \max \quad & \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big(\sum_{k\in\mathcal{K}_i} z_{jir}^k (\beta_i^k - \lambda_i^k) + \sum_{k\in\mathcal{K}_i^d} z_{jir}^k (\psi^k - \lambda_i^k) \Big) \cdot \tilde{\tau}_{jir} \\
& - \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1 (z_{ijr}^k - 1)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k) \\
& + \sum_{k\in\mathcal{K}} e^k \cdot (\gamma^k - \lambda_{o^k}^k) + \sum_{k\in\mathcal{K}} l^k \cdot (\lambda_{d^k}^k - \psi^k) \qquad (4.5)
\end{aligned}
$$

$$
\text{s.t.} \quad \beta_i^k - \beta_j^k - \eta_{ij}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{ijr}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{ijr}^k - \lambda_i^k + \lambda_j^k \le 0, \ \forall\ k \in \mathcal{K}, (i,j) \in \mathcal{A}, i \ne o^k, j \ne d^k, \qquad (4.6)
$$

$$
- \beta_j^k + \gamma^k - \eta_{o^k j}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{o^k jr}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{o^k jr}^k - \lambda_{o^k}^k + \lambda_j^k \le 0, \ \forall\ k \in \mathcal{K}, (o^k, j) \in \mathcal{A}, j \ne d^k, \qquad (4.7)
$$

$$
\beta_i^k - \psi^k - \eta_{id^k}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{id^k r}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{id^k r}^k - \lambda_i^k + \lambda_{d^k}^k \le 0, \ \forall\ k \in \mathcal{K}, (i, d^k) \in \mathcal{A}, i \ne o^k, \qquad (4.8)
$$

$$
\gamma^k - \psi^k - \eta_{o^k d^k}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{o^k d^k r}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{o^k d^k r}^k - \lambda_{o^k}^k + \lambda_{d^k}^k \le 0, \ \forall\ k \in \mathcal{K}, (o^k, d^k) \in \mathcal{A}, \qquad (4.9)
$$

$$
\sum_{k\in\mathcal{K}} \theta_{ijr}^k - \sum_{k\in\mathcal{K}} \xi_{ijr}^k \le 0, \ \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \qquad (4.10)
$$

$$\lambda_i^k \le h^k q^k,\ \forall\ i \in \mathcal{N}, k \in \mathcal{K}, \tag{4.11}$$

$$\psi^k - \lambda_{d^k}^k \le g^k,\ \forall\ k \in \mathcal{K}, \tag{4.12}$$

$$\boldsymbol{\beta} \ge \mathbf{0}, \boldsymbol{\gamma} \ge \mathbf{0}, \boldsymbol{\psi} \ge \mathbf{0}, \boldsymbol{\eta} \ge \mathbf{0}, \boldsymbol{\theta} \ge \mathbf{0}, \boldsymbol{\xi} \ge \mathbf{0}, \boldsymbol{\lambda} \ge \mathbf{0}, \tag{4.13}$$

where $\mathcal{K}_i = \{k \in \mathcal{K} : i \ne o^k \text{ and } i \ne d^k\}$ and $\mathcal{K}_i^d = \{k \in \mathcal{K} : i = d^k\}$.

Accordingly, we can use model DLP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$ to reformulate the inner minimization problem of the max-min model RP$(\boldsymbol{x}, \boldsymbol{z})$ of $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ defined by (3.29)–(3.40). This, together with the definitions of $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ and $\mathbb{U}(\Gamma)$ in (2.6) and (2.8), implies that the max-min model RP$(\boldsymbol{x}, \boldsymbol{z})$ for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ can then be reformulated to the following nonlinear optimization model with a bi-linear objective function:

$$\begin{aligned} F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \max\ & \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big(\sum_{k\in\mathcal{K}_i} z_{jir}^k (\beta_i^k - \lambda_i^k) + \sum_{k\in\mathcal{K}_i^d} z_{jir}^k (\psi^k - \lambda_i^k) \Big) \cdot \tilde{\tau}_{jir} \\ & - \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1 (z_{ijr}^k - 1)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k) \\ & + \sum_{k\in\mathcal{K}} e^k \cdot (\gamma^k - \lambda_{o^k}^k) + \sum_{k\in\mathcal{K}} l^k \cdot (\lambda_{d^k}^k - \psi^k) \qquad (4.14) \\ \text{s.t.}\ & (4.6) - (4.13), \qquad (4.15) \\ & \tilde{\tau}_{ijr} = \overline{\tau}_{ij} + \hat{\tau}_{ij} \delta_{ijr}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \qquad (4.16) \\ & -1 \le \delta_{ijr} \le 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \qquad (4.17) \\ & \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} |\delta_{ijr}| \le \Gamma. \qquad (4.18) \end{aligned}$$

Proposition 4.1 below indicates that the domain of each variable δ_{ijr} can be restricted to $\{-1, 0, 1\}$ without changing the optimal objective value of the nonlinear optimization model above.

Proposition 4.1 *There exists an optimal solution to the nonlinear optimization model defined in (4.14)–(4.18) such that $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$.*

Based on Proposition 4.1, we can now derive an MILP reformulation for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$, which is presented as follows in Proposition 4.2.

Proposition 4.2 *The max-min model RP$(\boldsymbol{x}, \boldsymbol{z})$ defined by (3.29)–(3.40) for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ can be equivalently written as the following maximization MILP model:*

$$F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \max \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \varphi_{jir} - \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k$$

$$+\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}[M_1(z^k_{ijr}-1)]\cdot(\theta^k_{ijr}+\xi^k_{ijr})$$
$$+\sum_{k\in\mathcal{K}}e^k\cdot(\gamma^k-\lambda^k_{o^k})+\sum_{k\in\mathcal{K}}l^k\cdot(\lambda^k_{d^k}-\psi^k) \tag{4.19}$$

s.t. *(4.6) – (4.13)*, (4.20)

$$\zeta_{ijr,-1}+\zeta_{ijr,0}+\zeta_{ijr,1}=1,\quad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{4.21}$$

$$\Big(\sum_{k\in\mathcal{K}_i}z^k_{jir}(\beta^k_i-\lambda^k_i)+\sum_{k\in\mathcal{K}^d_i}z^k_{jir}(\psi^k-\lambda^k_i)\Big)\tilde{\tau}_{jir,\ell}-M_2(1-\zeta_{jir,\ell})\le\varphi_{jir}$$
$$\le\Big(\sum_{k\in\mathcal{K}_i}z^k_{jir}(\beta^k_i-\lambda^k_i)+\sum_{k\in\mathcal{K}^d_i}z^k_{jir}(\psi^k-\lambda^k_i)\Big)\tilde{\tau}_{jir,\ell}+M_2(1-\zeta_{jir,\ell}),$$
$$\forall\ (j,i)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\},\ell\in\{-1,0,1\}, \tag{4.22}$$

$$\sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}(\zeta_{ijr,-1}+\zeta_{ijr,1})\le\Gamma. \tag{4.23}$$

$$\zeta_{ijr,\ell}\in\{0,1\},\quad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\},\ell\in\{-1,0,1\}. \tag{4.24}$$

An optimum solution to the maximization MILP model in Proposition 4.2 can be solved directly by an optimization solver. The objective value of the optimal solution obtained provides the worst-case second-stage cost $F_{RP}(\boldsymbol{x},\boldsymbol{z})$. The values of variables $\zeta_{ijr,-1}$ and $\zeta_{ijr,1}$ in the optimal solution for $(i,j)\in\mathcal{A}$ and $r\in\{1,...,|\mathcal{K}|\}$ can be used to compute the corresponding worst-case scenario $\boldsymbol{\delta}$, as shown below:

$$\delta_{ijr}=-\zeta_{ijr,-1}+\zeta_{ijr,1},\quad \forall\ (i,j)\in\mathcal{A}, r\in\{1,...,|\mathcal{K}|\}. \tag{4.25}$$

4.1.2. Algorithm Details In each iteration n, where $n=1,2,\cdots$, our RO-C&CG algorithm first solves model ROMILP(Λ), which is referred to as the master problem, for a particular subset Λ of $\mathbb{U}(\Gamma)$. Let $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}},\phi)$ indicate the optimal solution obtained for the master problem. Accordingly, $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})$ forms a nominal timely-implementable first-stage solution to model RO. For the first-stage solution $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})$, our RO-C&CG algorithm then solves the corresponding maximization MILP model defined by (4.19)–(4.24), which is referred to as the subproblem, to compute the worst-case second stage cost $F_{RP}(\hat{\boldsymbol{x}},\hat{\boldsymbol{z}})$ and to identify the corresponding worst-case scenario $\boldsymbol{\delta}^{(n)}$. Since ROMILP($\Lambda$) is a relaxation of model RO, its optimal objective value obtained is a lower bound on the optimal objective value of model RO. Since $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ forms a nominal timely-implementable first-stage solution to model RO, the sum of its first-stage total cost $(\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}}(c^k_{ij}q^k)\cdot\hat{x}^k_{ij}+\sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}f_{ij}\cdot\hat{y}_{ijr})$ and its second stage total cost $F_{RP}(\hat{\boldsymbol{x}},\hat{\boldsymbol{z}})$ provides an upper bound on the optimal objective value of model RO.

If the lower bound equals the upper bound, then model RO is solved to optimum, and our RO-C&CG algorithm terminates with an optimal solution given by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. Otherwise, it appends the identified worst-case scenario $\boldsymbol{\delta}^{(n)}$ to the subset Λ. As a result, model ROMILP(Λ) of the master problem is extended and strengthened with new decision variables $(\boldsymbol{v}^{(\boldsymbol{\delta})}, \boldsymbol{b}^{(\boldsymbol{\delta})}, \boldsymbol{w}^{(\boldsymbol{\delta})}, \boldsymbol{s}^{(\boldsymbol{\delta})})$ and their new constraints in (4.2)-(4.3). Our RO-C&CG algorithm then proceeds to the next iteration.

Here, we provide a summary of our RO-C&CG algorithm in Algorithm 1, along with its correctness and convergence in Theorem 4.1.

Algorithm 1 RO-C&CG Algorithm for Solving Model RO

1. Initially, set n to 1, and set the subset Λ of $\mathbb{U}(\Gamma)$ to $\{\mathbf{0}\}$.
2. Solve the master problem, i.e., model ROMILP(Λ), to obtain its optimal objective value denoted by LB and its optimal solution denoted by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \phi)$.
3. Solve the subproblem, i.e., the maximization MILP model defined by (4.19)–(4.24) for $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$, to obtain its optimal objective value that equals $F_{RP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$, and to compute a worst-case scenario $\boldsymbol{\delta}^{(n)}$ of $\boldsymbol{\delta}$ according to (4.25). Let UB denote the sum of $(\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}}(c_{ij}^k q^k)\cdot\hat{x}_{ij}^k + \sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|} f_{ij}\cdot\hat{y}_{ijr})$ and $F_{RP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$.
4. If $LB = UB$, then the algorithm terminates and returns an optimal solution given by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. Otherwise, update $\Lambda = \Lambda \bigcup \{\boldsymbol{\delta}^{(n)}\}$, update $n = n+1$, and go to Step 2 for the next iteration.

Theorem 4.1 *Algorithm 1 terminates in a finite number of iterations and returns an optimal solution for model RO.*

4.2. C&CG Algorithm for Robust Satisficing Model

For the sake of clarity, for any given first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ we define $F_1(\boldsymbol{x}, \boldsymbol{y})$ as follows to represent its first-stage total cost:

$$F_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}}(c_{ij}^k q^k)\cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|} f_{ij}\cdot y_{ijr} \tag{4.26}$$

For any possible scenario $\boldsymbol{\delta}\in\mathbb{U}$, $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ defined in (3.41)-(3.42) represents the second-stage cost of any first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ under $\boldsymbol{\delta}$. Accordingly, model RS proposed in Section 3.2.2 can be rewritten as the following noncompact MILP, referred to as model RSMILP:

$$\begin{aligned}
[\text{RSMILP}]\ \min\ & \rho \\
\text{s.t. } & F_1(\boldsymbol{x}, \boldsymbol{y}) + \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}}(h^k q^k)\cdot w_i^{k(\boldsymbol{\delta})} + \sum_{k\in\mathcal{K}} g^k\cdot s^{k(\boldsymbol{\delta})} - \mathcal{Z} \le \rho\|\boldsymbol{\delta}\|_1, \quad \forall\ \boldsymbol{\delta}\in\mathbb{U}, && (4.27)\\
& (\boldsymbol{v}^{(\boldsymbol{\delta})}, \boldsymbol{b}^{(\boldsymbol{\delta})}, \boldsymbol{w}^{(\boldsymbol{\delta})}, \boldsymbol{s}^{(\boldsymbol{\delta})}) \in \mathcal{Q}(\boldsymbol{\delta}), \quad \forall\ \boldsymbol{\delta}\in\mathbb{U}, && (4.28)\\
& \rho \ge 0, (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}}) \in \mathcal{X}.
\end{aligned}$$

Here, $(\boldsymbol{v}^{(\boldsymbol{\delta})}, \boldsymbol{b}^{(\boldsymbol{\delta})}, \boldsymbol{w}^{(\boldsymbol{\delta})}, \boldsymbol{s}^{(\boldsymbol{\delta})})$ represents a vector of second-stage decision variables associated with each possible scenario $\boldsymbol{\delta}$ in $\mathbb{U}$, similar to those in model ROMILP. Constraints (4.27) and (4.28) ensure that the deviation of the total two-stage cost from the prescribed target $\mathcal{Z}$ does not exceed $\rho\|\boldsymbol{\delta}\|_1$ for every possible scenario $\boldsymbol{\delta} \in \mathbb{U}$. As a result, solving model RS is reduced to solving the above noncompact MILP model RSMILP.

Model RSMILP can also be relaxed by replacing $\mathbb{U}$ in constraints (4.27) and (4.28) with any of its subsets $\Lambda \subseteq \mathbb{U}$. The resulting relaxation is referred to as model RSMILP(Λ). The relaxation can also be strengthened by appending to Λ more possible scenarios $\boldsymbol{\delta}$ in $\mathbb{U}$. When Λ equals $\mathbb{U}$, model RSMILP(Λ) and model RSMILP are equivalent.

Accordingly, similar to the RO-C&CG algorithm for model RO, our C&CG algorithm for model RS, which is referred to as the RS-C&CG algorithm, iteratively solves model RSMILP(Λ) to obtain a first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and append the corresponding worst-case scenario $\boldsymbol{\delta}$ to Λ, until $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ implies an optimal solution.

However, compared with the RO-C&CG algorithm for model RO, the RS-C&CG algorithm for model RS faces a more significant challenge in computing a worst-case scenario $\boldsymbol{\delta}$ in each iteration. This is mainly because such a worst-case scenario $\boldsymbol{\delta}$ with respect to model RS (to be defined later) is different from that with respect to model RO, and computing it requires fractional optimization, which is complicated. To tackle this challenge, we develop a bisection search procedure, which is illustrated below and is followed by the details of our RS-C&CG algorithm for model RS.

4.2.1. Computing the worst-case scenario $\boldsymbol{\delta}$ by enhanced bisection search Consider any given first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. For any possible scenario $\boldsymbol{\delta} \in \mathbb{U}$, the ratio $(F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z})/\|\boldsymbol{\delta}\|_1$ represents a *normalized cost deviation* from the prescribed target under scenario $\boldsymbol{\delta}$. For simplicity of the presentation, here we slightly abuse the notation to define that $\sigma/\|\mathbf{0}\|_1 = 0$ for $\sigma = 0$, $\sigma/\|\mathbf{0}\|_1 = +\infty$ for $\sigma > 0$, and $\sigma/\|\mathbf{0}\|_1 = -\infty$ for $\sigma < 0$. Accordingly, constraints (4.27) and (4.28) in model RSMILP imply that the normalized cost deviation with respect to the prescribed target $\mathcal{Z}$ cannot exceed ρ for all $\boldsymbol{\delta} \in \mathbb{U}$.

The maximum value of the normalized cost deviation over all $\boldsymbol{\delta} \in \mathbb{U}$ is defined as the *worst-case normalized cost deviation* of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and the corresponding $\boldsymbol{\delta}$ that leads to the ratio achieving the maximum value is referred to as the *worst-case scenario* for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, with respect to model RS. Computing such a worst-case scenario $\boldsymbol{\delta}$ can be formulated as the following fractional optimization model, which is referred to as model FO$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$:

$$[\mathrm{FO}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})] \ \max_{\boldsymbol{\delta} \in \mathbb{U}} \ \frac{F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}}{\|\boldsymbol{\delta}\|_1}. \tag{4.29}$$

To solve model FO($\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$), we develop an enhanced bisection search procedure as follows. Let $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ indicate the optimal objective value of model FO($\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$). Our enhanced bisection search procedure starts with a lower bound ρ_l and an upper bound ρ_h on the value of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. In each iteration, it first evaluates whether the middle point $\hat{\rho} = (\rho_l + \rho_u)/2$ is larger than $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, or not. If $\hat{\rho}$ is larger than $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, the upper bound ρ_h is decreased to $\hat{\rho}$. Otherwise, the lower bound ρ_l is increased to $\hat{\rho}$. It then further enhances the lower bound ρ_l so that the enhanced ρ_l equals the normalized cost deviation under some $\boldsymbol{\delta}$. The procedure terminates if ρ_l is proved to be equal to $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$; otherwise, it proceeds to the next iteration.

Given any guessed value $\hat{\rho}$, consider the following optimization model, which does not involve fractional optimization and whose optimal objective value is denoted by $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho})$:

$$G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) = \max_{\boldsymbol{\delta} \in \mathbb{U}} \; F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \hat{\rho}\|\boldsymbol{\delta}\|_1. \tag{4.30}$$

Lemma 4.2 below indicates that one can determine whether $\hat{\rho}$ is less than, greater than, or equal to $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ by evaluating the value of $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho})$. It also implies that $\hat{\rho}$ is proved to be equal to $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ if $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) = 0$.

Lemma 4.2 *If $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) > 0$, then $\hat{\rho} < \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Otherwise, if $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) \leq 0$, then $\hat{\rho} \geq \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. If $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) = 0$, then $\hat{\rho} = \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.*

Recall that $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ is defined by a linear program, which, according to Lemma 4.1 always has a feasible solution. Similar to our reformulation of $F_{\mathrm{RP}}(\boldsymbol{x}, \boldsymbol{z})$, we can replace the LP formulation of $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ with its dual to reformulate the model defined in (4.30) for $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho})$ as the following nonlinear optimization model in (4.31)–(4.34), and further reformulate it as an MILP.

$$\begin{aligned} G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) = \max \; & F_1(\boldsymbol{x}, \boldsymbol{y}) + \Big[\sum_{(j,i) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big(\sum_{k \in \mathcal{K}_i} z_{jir}^k (\beta_i^k - \lambda_i^k) + \sum_{k \in \mathcal{K}_i^d} z_{jir}^k (\psi^k - \lambda_i^k) \Big) \cdot \tilde{\tau}_{jir} \\ & - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k + \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1 (z_{ijr}^k - 1)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k) \\ & + \sum_{k \in \mathcal{K}} e^k \cdot (\gamma^k - \lambda_{o^k}^k) + \sum_{k \in \mathcal{K}} l^k \cdot (\lambda_{d^k}^k - \psi^k) \Big] - \mathcal{Z} - \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \hat{\rho} |\delta_{ijr}| \qquad (4.31) \\ \text{s.t.} \quad & (4.6) - (4.13) \qquad (4.32) \\ & \tilde{\tau}_{ijr} = \overline{\tau}_{ij} + \hat{\tau}_{ij} \delta_{ijr}, \quad \forall \; (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \qquad (4.33) \\ & -1 \leq \delta_{ijr} \leq 1, \quad \forall \; (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}. \qquad (4.34) \end{aligned}$$

To achieve this, we first need to establish Proposition 4.3 below, which states that the domain of each variable δ_{ijr} can be restricted to $\{-1, 0, 1\}$ without changing the optimal objective value of

the nonlinear optimization model above. We can then establish Proposition 4.2 to obtain the MILP reformulation for $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho})$.

Proposition 4.3 *There exists an optimal solution to the nonlinear optimization model defined in (4.31)–(4.34) such that $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$.*

Proposition 4.4 *The nonlinear model defined in* (4.30) *for $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho})$ can be equivalently written as the following MILP:*

$$\max \quad F_1(\boldsymbol{x}, \boldsymbol{y}) - \mathcal{Z} + \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \hat{\varphi}_{jir} - \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1(z_{ijr}^k - 1)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k)$$
$$+ \sum_{k\in\mathcal{K}} e^k \cdot (\gamma^k - \lambda_{o^k}^k) + \sum_{k\in\mathcal{K}} l^k \cdot (\lambda_{d^k}^k - \psi^k) \tag{4.35}$$

$$s.t. \quad (4.6) - (4.13), \tag{4.36}$$

$$\hat{\zeta}_{ijr,-1} + \hat{\zeta}_{ijr,1} + \hat{\zeta}_{ijr,0} = 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{4.37}$$

$$\Big(\sum_{k\in\mathcal{K}_i} z_{jir}^k (\beta_i^k - \lambda_i^k) + \sum_{k\in\mathcal{K}_i^d} z_{jir}^k (\psi^k - \lambda_i^k) \Big) \tilde{\tau}_{jir,\ell} - \hat{\rho}|\ell| - M_3(1 - \hat{\zeta}_{jir,\ell}) \le \hat{\varphi}_{jir}$$
$$\le \Big(\sum_{k\in\mathcal{K}_i} z_{jir}^k (\beta_i^k - \lambda_i^k) + \sum_{k\in\mathcal{K}_i^d} z_{jir}^k (\psi^k - \lambda_i^k) \Big) \tilde{\tau}_{jir,\ell} - \hat{\rho}|\ell| + M_3(1 - \hat{\zeta}_{jir,\ell}),$$
$$\forall\ (j,i) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \ell \in \{-1, 0, 1\}, \tag{4.38}$$

$$\hat{\zeta}_{ijr,\ell} \in \{0, 1\}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \ell \in \{-1, 0, 1\}. \tag{4.39}$$

As shown below, similar to our MILP reformulation of $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ shown in Proposition 4.2, we can also utilize the values of variables $\hat{\zeta}_{ijr,-1}$ and $\hat{\zeta}_{ijr,1}$, for $(i,j) \in \mathcal{A}$ and $r \in \{1, ..., |\mathcal{K}|\}$, in the optimal solution of the MILP formulation in Proposition 4.4, to further compute the corresponding worst-case scenario $\boldsymbol{\delta}$:

$$\delta_{ijr} = -\hat{\zeta}_{ijr,-1} + \hat{\zeta}_{ijr,1}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}. \tag{4.40}$$

Next, we can establish Lemma 4.3 below, which enables us to set the initial values for the lower bound ρ_l and the upper bound ρ_h on $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ in our enhanced bisection search procedure.

Lemma 4.3

1. *If $F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\mathbf{0})) - \mathcal{Z} > 0$, then $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = +\infty$;*
2. *Otherwise, $(F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}_l)) - \mathcal{Z})/\|\boldsymbol{\delta}_l\|_1 \le \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ for each $\boldsymbol{\delta}_l \in \mathbb{U} \setminus \{\mathbf{0}\}$, and $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \le \max\{0, F_1(\boldsymbol{x}, \boldsymbol{y}) + \max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))\} - \mathcal{Z}\}$.*

Since $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ is defined by a linear program, it can be obtained directly by an optimization solver. Model $\max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))\}$ is equivalent to $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ defined in (3.29)–(3.40) with $\mathbb{U}(\Gamma)$ being relaxed to $\mathbb{U}$ (i.e., with $\Gamma = +\infty$), which can be transformed to an MILP as shown in (4.19)–(4.24). Thus, it can also be solved by an optimization solver.

According to Lemma 4.3, if $(F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{0})) - \mathcal{Z}) > 0$, then the worst-case normalized cost deviation $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = +\infty$ and $\boldsymbol{0}$ is the worst-case scenario for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Otherwise, we know that $(F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{0})) - \mathcal{Z})/\|\boldsymbol{\delta}\|_1$ for any $\boldsymbol{\delta} \in \mathbb{U} \setminus \{\boldsymbol{0}\}$ and $\max\{0, F_1(\boldsymbol{x}, \boldsymbol{y}) + \max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))\} - \mathcal{Z}\}$ provide a lower bound and an upper bound on the worst-case normalized cost deviation $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, respectively.

Moreover, consider any given lower bound ρ_l, which is known to not exceed $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Let $\boldsymbol{\delta}(\rho_l)$ indicate the realization of $\boldsymbol{\delta}$, derived by (4.40) from the optimal solution to model $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho_l)$ defined in (4.35)–(4.39). Define ρ_l' below to indicate the normalized cost deviation under $\boldsymbol{\delta}(\rho_l)$

$$\rho_l' = \frac{F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z}}{\|\boldsymbol{\delta}(\rho_l)\|_1}. \tag{4.41}$$

Lemma 4.4 below indicates that if ρ_l is a lower bound on $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, then ρ_l' is a lower bound on $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ that is greater than or equal to ρ_l. As a result, in each iteration of our enhanced bisection search procedure, the lower bound ρ_l is enhanced to ρ_l'.

Lemma 4.4 *If $\rho_l \le \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, then ρ_l' defined in (4.41) satisfies that $\rho_l \le \rho_l' \le \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.*

Below, we provide a summary of our enhanced bisection search procedure for any given first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ in Algorithm 2, along with its correctness and convergence in Theorem 4.2.

Algorithm 2 An Enhanced Bisection Search Procedure for Any Given $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$

1. If $(F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{0})) - \mathcal{Z}) > 0$, return $+\infty$ as the value of the worst-case normalized cost deviation of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and $\boldsymbol{0}$ as the worst-case scenario for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.
2. Initially, choose any $\boldsymbol{\delta}_l \in \mathbb{U} \setminus \{\boldsymbol{0}\}$, set $\rho_l = (F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}_l)) - \mathcal{Z})/\|\boldsymbol{\delta}_l\|_1$, and set $\rho_h = \max\{0, F_1(\boldsymbol{x}, \boldsymbol{y}) + \max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))\} - \mathcal{Z}\}$.
3. Set $\hat{\rho} = (\rho_h + \rho_l)/2$, solve the maximization MILP model defined in (4.35)–(4.39) to compute $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho})$.
4. If $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) > 0$, increase ρ_l to $\hat{\rho}$, and if $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) < 0$, decrease ρ_h to $\hat{\rho}$. Then go to Step 5. However, if $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) = 0$, increase ρ_l to $\hat{\rho}$, derive the scenario $\boldsymbol{\delta}(\rho_l)$ from the optimal solution to the model by (4.40), and then go to Step 6.

5. Enhancement: Solve the maximization MILP model defined in (4.35)–(4.39) to compute $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho_l)$, derive the scenario $\boldsymbol{\delta}(\rho_l)$ from the optimal solution to the model by (4.40), and compute ρ_l' from $\boldsymbol{\delta}(\rho_l)$ by (4.41). If $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho_l) = 0$, then go to Step 6. Otherwise, set ρ_l to ρ_l', and go to Step 3 for the next iteration.
6. Return ρ_l as the worst-case normalized cost deviation $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and return $\boldsymbol{\delta}(\rho_l)$ as the worst-case scenario for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.

Theorem 4.2 *Consider any given feasible first-stage decisions* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.

1. *Algorithm 2 is guaranteed to terminate within a finite number of iterations, with the value of* $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ *and a worst-case scenario* $\boldsymbol{\delta}$ *for* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ *returned.*
2. *Let* $\rho_l^{(0)}$ *and* $\rho_h^{(0)}$ *denote the initial values of* ρ_l *and* ρ_h *assigned in Step 3 of Algorithm 2. Then, for any* $\epsilon > 0$*, after* $\lceil \log_2((\rho_h^{(0)} - \rho_l^{(0)})/\epsilon) \rceil$ *iterations of Steps 3–6, Algorithm 2 obtains a lower bound* ρ_l *on* $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ *and a scenario* $\boldsymbol{\delta}(\rho_l) \in \mathbb{U}$*, satisfying that* $\rho_l \leq \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \leq \rho_l + \epsilon$ *and that* $F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z} \geq \rho_l \|\boldsymbol{\delta}(\rho_l)\|_1$.

Theorem 4.2 indicates that the enhanced bisection search procedure in Algorithm 2 is an exact algorithm that solves model FO$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ within a finite number of iterations. Theorem 4.2 also implies that Algorithm 2 solves model FO$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ to an accuracy $\epsilon > 0$ within $\lceil \log_2((\rho_h^{(0)} - \rho_l^{(0)})/\epsilon) \rceil$ iterations. As a result, if one terminates Algorithm 2 after $\lceil \log_2((\rho_h^{(0)} - \rho_l^{(0)})/\epsilon) \rceil$ iterations, the latest values of ρ_l and $\boldsymbol{\delta}(\rho_l)$ can be obtained as output. These values guarantee that ρ_l and $\rho_l + \epsilon$ respectively provide a lower bound and an upper bound on the worst-case normalized cost deviation $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and that constraint (4.27) is violated by $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ under $\boldsymbol{\delta}(\rho_l)$ for all $\rho < \rho_l$.

It is also worth noting that Step 5 is essential to guarantee that Algorithm 2 solves model FO$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ within a finite number of iterations. Without the enhancement in Step 5, Algorithm 2 functions as a standard bisection search procedure. In this standard procedure, valid lower and upper bounds of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ can be obtained with their gap smaller than a given tolerance $\epsilon > 0$ within a finite number of iterations. However, this standard bisection search procedure does not guarantee to produce the exact value of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ within a finite number of iterations. Alternatively, one can apply only Step 5 of Algorithm 2, but iteratively, to acquire the exact value of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ within a finite number of iterations. However, this approach does not ensure producing a valid upper bound on $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ until the exact value of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is reached. To overcome these limitations, we introduce the utilization of Step 5 in Algorithm 2 to enhance the standard bisection search procedure. As a result, our enhanced standard bisection search procedure guarantees to produce

valid lower and upper bounds of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ in each iteration, as stated in the second statement of Theorem 4.2. It also ensures producing the exact value of $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ within a finite number of iterations, as stated in the first statement of Theorem 4.2.

4.2.2. Algorithm Details Our RS-C&CG algorithm also follows the C&CG framework. In each iteration n, where $n = 1, 2, \cdots$, it first solves model RSMILP(Λ) as the master problem for a particular subset Λ of $\mathbb{U}$. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \phi)$ indicate the optimal solution obtained for the master problem. Accordingly, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ forms a nominal timely-implementable first-stage solution to model RS. For the first-stage solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, our RS-C&CG algorithm then applies the enhanced bisection search procedure in Algorithm 2 to solve the fractional optimization model FO$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ as the subproblem, and obtain the worst-case normalized cost deviation $\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, denoted by $\rho^{(n)}$, as well as the corresponding worst-case scenario, denoted by $\boldsymbol{\delta}^{(n)}$. Since RSMILP($\Lambda$) is a relaxation of model RS, its optimal objective value obtained is a lower bound on the optimal objective value of model RS. Since $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ forms a nominal timely-implementable first-stage solution to model RS, it can be seen that the positive part of the worst-case normalized cost deviation, denoted by $\max\{0, \rho^{(n)}\}$, provides an upper bound on the optimal objective value of model RS.

If the lower bound equals the upper bound, then model RS is solved to optimum, and our RS-C&CG algorithm terminates with an optimal solution given by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. Otherwise, it appends the identified scenario $\boldsymbol{\delta}^{(n)}$ to the subset Λ. As a result, model RSMILP(Λ) of the master problem is extended and strengthened with new decision variables $(\boldsymbol{v}^{(\boldsymbol{\delta})}, \boldsymbol{b}^{(\boldsymbol{\delta})}, \boldsymbol{w}^{(\boldsymbol{\delta})}, \boldsymbol{s}^{(\boldsymbol{\delta})})$ and their new constraints in (4.27)- (4.28). Our RS-C&CG algorithm then proceeds to the next iteration.

Here, we provide a summary of our RS-C&CG algorithm in Algorithm 3, along with its correctness and convergence in Theorem 4.3.

Algorithm 3 RS-C&CG Algorithm for Solving Model RS

1. Initially, set the iteration number n to 1, and set the subset Λ of $\mathbb{U}$ to $\{\mathbf{0}\}$.
2. Solve the master problem, i.e., model RSMILP(Λ), to obtain its optimal objective value denoted by LB and its optimal solution denoted by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \phi)$.
3. Apply the enhanced bisection search procedure in Algorithm 2 to solve the subproblem FO$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, so as to obtain the worst-case normalized cost deviation $\rho^*(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, denoted by $\rho^{(n)}$, and to obtain the corresponding worst-case scenario, denoted by $\boldsymbol{\delta}^{(n)}$. Let UB denote $\max\{0, \rho^{(n)}\}$.
4. If $LB = UB$, then the algorithm terminates and returns an optimal solution given by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. Otherwise, update $\Lambda = \Lambda \bigcup \{\boldsymbol{\delta}^{(n)}\}$, update $n = n + 1$, and go to Step 2 for the next iteration.

Theorem 4.3 *Algorithm 3 terminates in a finite number of iterations and returns an optimal solution to model RS.*

5. Computational Experiments

We performed two sets of computational experiments using instances randomly generated in accordance with the approach outlined in Section 5.1.

- The first set of experiments aimed to assess the performance of our exact algorithms in solving model RO and model RS of the robust CTSNDP with uncertain travel times. Results of the first set of experiments, reported in Section 5.2, demonstrate the effectiveness of our exact algorithms and tractability of our proposed formulations.
- The second set of experiments aimed to evaluate the quality of the solutions obtained from model RO and model RS with different parameters defining the uncertainty sets, by comparing their total costs in the nominal scenario, as well as their total costs in the worst-case scenario and on average over randomly generated scenarios. Results of the second set of experiments, reported in Section 5.3, demonstrate the robustness of the solutions obtained from model RO and model RS, along with their corresponding prices of robustness.

We implemented our RO-C&CG algorithm for model RO and RS-C&CG algorithm for model RS in Java, utilizing the Gurobi solver (v.10.0.2) to solve the corresponding master problems and subproblems. All experiments were conducted on a 64-bit Windows 10 operating system, using a PC equipped with an Intel(R) Core(TM) i7-8700 CPU clocked at 3.20 GHz and 64 GB RAM.

5.1. Instance Generation and Parameter Setting

For our experiments, we generated test instances of the robust CTSNDP based on the 7 instance classes (named R4-R10) of the fixed-charge capacitated multi-commodity network design (CMND) problem available in the literature (Ghamlouche et al. 2003). These classes of CMND instances have been utilized in previous studies to generate test instances for various stochastic capacitated fixed charge network design problems (Crainic et al. 2011, Sarayloo et al. 2021a,b). As summarized in columns $|\mathcal{N}|$, $|\mathcal{A}|$, and $|\mathcal{K}|$ of Table 5.1, the sizes of the node set $\mathcal{N}$, arc set $\mathcal{A}$, and commodity set $\mathcal{K}$ vary from 10 to 20, from 60 to 120, and from 10 to 50, respectively, among instances belonging to different classes.

For each CMND instance of the 7 classes, we first generated fixed costs and time attributes (including nominal travel times, commodities' earliest available times, and due times) of the CTSNDP following an approach similar to that presented in Boland et al. (2017). We then generated

Table 5.1 Computational Performance of RO-C&CG and RS-C&CG Algorithms.

Class	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{K}\|$	RO-C&CG					RS-C&CG				
				opt%	g%	T	Im%		opt%	g%	T	Im%	
							mean	max				mean	max
R4	10	60	10	100.0	0.0	0.2	6.4	29.7	100.0	0.0	0.3	37.8	100.0
R5	10	60	25	100.0	0.0	7.3	7.2	28.1	100.0	0.0	8.1	46.7	100.0
R6	10	60	50	73.3	0.4	9672.7	6.5	21.5	66.7	13.7	11520.1	46.1	85.5
R7	10	82	10	100.0	0.0	0.4	9.7	25.8	100.0	0.0	0.4	22.2	100.0
R8	10	83	25	100.0	0.0	11.7	11.8	23.1	100.0	0.0	12.3	48.7	85.4
R9	10	83	50	86.7	0.1	7117.9	6.9	11.9	100.0	0.0	1530.5	37.4	82.2
R10	20	120	40	100.0	0.0	489.5	9.0	26.7	90.0	0.5	4402.3	52.1	91.9
Mean				94.3	0.1	2471.4	8.2	29.7	93.8	1.9	2486.3	41.6	100.0

maximum deviations of travel times for arcs, unit in-storage holding costs, and unit delay penalties for commodities. Accordingly, we obtained in total 210 test instances for our experiments with 30 instances for each class (see Appendix D for their detailed parameter settings).

For our experiments, we set the parameters of our model RO and model RS as follows. The uncertainty budget Γ of model RO was set to $\lceil \mu_\Gamma \cdot |\mathcal{K}| \rceil$. The cost target $\mathcal{Z}$ of model RS was set to $\lceil (1+\mu_z) \cdot \mathcal{Z}_0 \rceil$. We chose different values of the coefficients μ_Γ and μ_z in different sets of experiments, which will be explained in Section 5.2 and Section 5.3, respectively.

5.2. Performance of RO-C&CG and RS-C&CG Algorithms

In the first set of experiments, we evaluated the computational performance of the RO-C&CG and the RS-C&CG algorithms over the 210 test instances of the robust CTSNDP. Each algorithm terminates and returns the best upper and lower bounds found when its running time exceeds an 8-hour time limit or when the optimality gap between its best upper and lower bounds found is below a 0.01% threshold. Moreover, as a benchmark, for each test instance we solved the deterministic model DO directly by the Gurobi solver. The optimal objective value of the deterministic model DO is used as the cost target $\mathcal{Z}_0$ in the RS-C&CG algorithm. Additionally, we set $\mu_\Gamma = \mu_z = 0.05$.

The computational results are presented in Table 5.1. For each class of the test instances and for each algorithm, we report the percentage of the instances solved to optimality in column opt%, the average optimality gap in column g% (defined as the percentage gap between the best upper and lower bounds found), and the average computational time in CPU seconds in column T.

Let UB^{RO} and UB^{RS} indicate the best upper bound values obtained in the RO-C&CG and RS-C&CG algorithms for model RO and model RS, respectively. For the optimal solution $(\boldsymbol{x}_d, \boldsymbol{y}_d, \boldsymbol{z}_d)$ obtained from the deterministic model DO, we can follow the approach in Section 4.1 to solve $F_{RP}(\boldsymbol{x}_d, \boldsymbol{z}_d)$ so as to compute the objective value of $(\boldsymbol{x}_d, \boldsymbol{y}_d, \boldsymbol{z}_d)$ in model RO, denoted

by UB_d^{RO}. We can also follow the approach in Section 4.2 to solve $\rho(\boldsymbol{x}_d, \boldsymbol{y}_d, \boldsymbol{z}_d)$ so as to compute the objective value of $(\boldsymbol{x}_d, \boldsymbol{y}_d, \boldsymbol{z}_d)$ in model RS, denoted by UB_d^{RS}. To evaluate the improvements made by UB^{RO} and UB^{RS} against the benchmark values UB_d^{RO} and UB_d^{RS}, we compute their improvement percentages, which are defined as ratios $\frac{(UB_d^{RO} - UB^{RO})}{UB_d^{RO}} \times 100\%$ and $\frac{(UB_d^{RS} - UB^{RS})}{UB_d^{RS}} \times 100\%$, respectively. Their mean and maximum values for each class of instances are shown in columns Im% of Table 5.1.

The results in Table 5.1 confirm the effectiveness of our RO-C&CG and RS-C&CG algorithms in solving model RO and model RS. Within the time limit, both algorithms can solve around 94% of all 210 instances to exact optimality, as well as achieve optimality gaps of 0.1% and 2.6% on average, respectively. Moreover, as shown in columns Im%, compared with the deterministic optimal solutions, the best upper bounds produced by our RO-C&CG and RS-C&CG algorithms significantly improve the objective values with respect to model RO and model RS by 8.2% and 41.6% on average and by 29.7% and 100.0% at maximum, respectively.

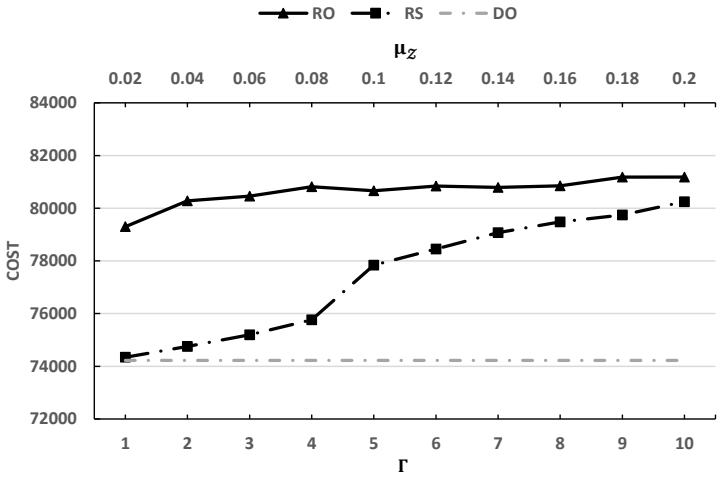
Table 5.1 also demonstrates the comparable computational performance of the two C&CG-based algorithms in solving model RO and model RS, respectively, as evidenced by the columns opt%, g%, and T. These results confirm the computational tractability of both model RO and model RS that we derive for the robust CTSNDP, underscoring their practical usefulness. Notably, our study presents the first development of a C&CG-based algorithm for effectively solving a two-stage robust satisficing model with a polyhedral uncertainty set. The encouraging outcomes motivate further exploration of its potential extensions to other optimization problems that encompass uncertainties.

5.3. Solution Qualities of Models RO and RS

Models RO and RS have distinct objectives for achieving robustness. Following the approach in Bertsimas and Sim (2004) and Atamtürk and Zhang (2007), our second set of experiments compares the first-stage solutions obtained from each model under different criteria. Our aim is to evaluate their robustness and the associated trade-offs, or price of robustness. Specifically, we compare their total costs in the nominal scenario to that of the optimal nominal solutions, and then compare their worst-case and average total costs over randomly generated scenarios to the solutions obtained from a stochastic programming model (to be detailed in Section 5.3.2).

Our second set of experiments focuses on instances in class R7, as these instances were all solved to optimality for models RO, RS, DO, and the stochastic programming model. For each instance in R7, we used our RO-C&CG algorithm to solve model RO for each uncertainty budget $\mathbb{U}(\Gamma)$ with $\Gamma \in \{1, 2, ..., 10\}$, applied our RS-C&CG algorithm to solve model RS for each cost target $\mathcal{Z} = \lceil (1 + \mu_z) \cdot \mathcal{Z}_0 \rceil$ with $\mu_z \in \{0.02, 0.04, ..., 0.2\}$, and utilized the Gurobi solver to solve model

Figure 2 Comparing the total costs of solutions obtained from models RO, RS, and DO in the nominal scenario.



DO and the stochastic programming model. Recall that $\mathcal{Z}_0$ is the optimal objective value of the deterministic model DO.

5.3.1. Nominal Scenario Under different parameters Γ and μ_z, we first compare the total costs of the solutions obtained from models RO and RS in nominal scenarios to that of the optimal nominal solution. These evaluations were conducted across 30 instances of class R7, and the results are presented in Figure 2, where the total cost along the vertical axis is the mean across all instances.

As shown in Figure 2, the solutions obtained from model RS exhibit better overall performance in the nominal scenario than those from model RO. Increasing the value of μ_z from 0.02 to 0.2, which increases the cost target, the total nominal cost of the solutions from model RS gradually increases, causing its gap from that of the optimal nominal solution to increase from 0.2% to 8.1%. This suggests that μ_z is effective in controlling the price of robustness for solutions of model RS in terms of performance in the nominal scenario.

In contrast, as the value of Γ increases and the budgeted uncertainty expands, only when Γ is small, the total nominal cost of solutions obtained from model RO exhibits an increasing trend in the gap from the optimal nominal solution, varying from 6.8% to 8.2% for Γ increasing from 1 to 2. When Γ is large, increasing its value only slightly changes the total nominal cost of solutions obtained from model RO, and the change is not always positive. These results suggest that Γ cannot be effectively used to control the price of robustness for solutions of model RO in terms of performance in the nominal scenario.

Therefore, if the decision maker places a high value on nominal performance, model RS is a suitable choice as its parameter μ_z allows the decision maker to adjust the trade-off between a solution's robustness and nominal performance. This provides the decision maker with greater

control over the price of robustness and allows them to make more informed decisions based on their priorities.

5.3.2. Randomly Generated Scenarios Under different parameters Γ and μ_z, we next evaluate the worst-case and average total costs of solutions obtained from models RO and RS over randomly generated scenarios for instances in class R7. We compare them to those of the optimal solution of a stochastic programming (SP) model. The SP model aims to minimize the expected total cost over all possible scenarios in a given scenario set Π. The probability of each scenario $\boldsymbol{\delta} \in \Pi$ is represented by $Prob(\boldsymbol{\delta})$. Model SP can thus be formulated as follows:

$$\begin{aligned} [\text{SP}] \quad \min \ & \sum_{\boldsymbol{\delta}\in\Pi} Prob(\boldsymbol{\delta}) \Big(\sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} y_{ijr} + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) \Big) \\ s.t. \ & (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}}) \in \mathcal{X}. \end{aligned}$$

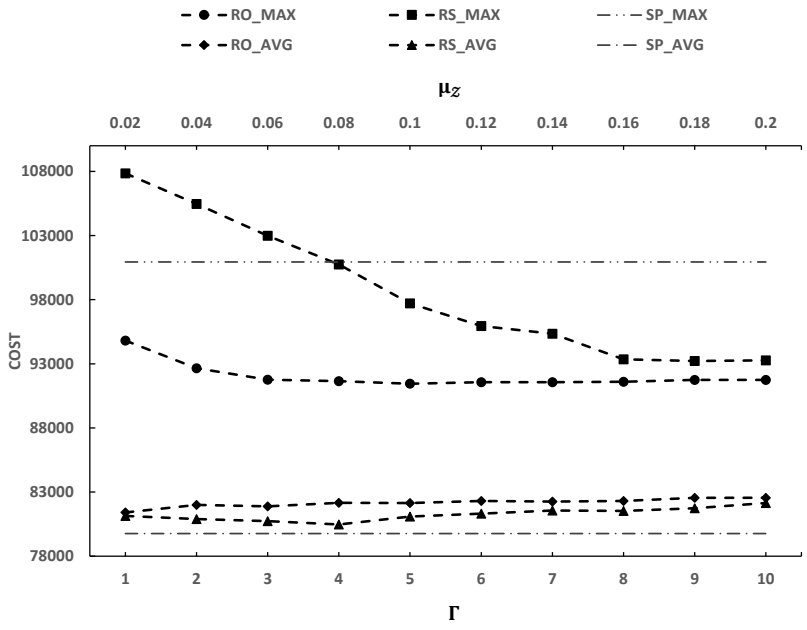
Since the second-stage cost $F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ is defined by a minimization LP, it can be seen that model SP is a minimization MILP, which can be directly solved by an optimization solver.

In class R7, we generated 200 scenarios at random to create the set Π for model SP for each of the 30 instances. We then used the Gurobi solver to solve model SP over these scenarios. To generate each scenario $\boldsymbol{\delta}$, we drew each of the realization of δ_{ijr} for $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$ uniformly from the set $\{-1, -(\hat{\tau}_{ij}-1)/\hat{\tau}_{ij}, -(\hat{\tau}_{ij}-2)/\hat{\tau}_{ij}, \cdots, -1/\hat{\tau}_{ij}, 0, 1/\hat{\tau}_{ij}, \cdots, (\hat{\tau}_{ij}-1)/\hat{\tau}_{ij}, 1\}$. We set $Prob(\boldsymbol{\delta})$ to $1/200$ so that each possible scenario has equal probability. This ensures that each travel time $\tilde{\tau}_{ijr}(\boldsymbol{\delta}) \in \mathbb{N}_{>0}$ for $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$ was uniformly distributed in interval $\{\overline{\tau}_{ij} - \hat{\tau}_{ij}, \overline{\tau}_{ij} - \hat{\tau}_{ij} + 1, \overline{\tau}_{ij} - \hat{\tau}_{ij} + 2, \cdots, \overline{\tau}_{ij} + \hat{\tau}_{ij}\}$.

Using the same approach, we generated 1000 random *testing scenarios* for each of the 30 instances in class R7. For each instance, we compared the solutions obtained by models RO, RS, and SP based on their worst-case performance and average performance over all testing scenarios. For each solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ obtained from models RO, RS, and SP, we computed its total cost $\sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} y_{ijr} + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$ for each testing scenario $\boldsymbol{\delta}$. Among these costs of all the scenarios, we then computed their maximum and average values to measure the worst-case and the average performances of the solution, respectively. The results are presented in Figure 3, where the total cost along the vertical axis is the mean across all instances, and the legends denote MAX and AVG for the worst-case and average costs, respectively.

From RO_MAX, RS_MAX, and SP_MAX in Figure 3, we can observe that the solutions produced by model RO demonstrate the highest level of robustness against travel time uncertainty. This is reflected in their superior worst-case performance, which outperforms the solutions produced by

Figure 3 **Comparing the worst-case and the average total costs of solutions obtained from models RO, RS, and SP over randomly generated scenarios.**



models RS and SP by approximately $1 \sim 12\%$. Although the worst-case performance of solutions obtained from model RS is worse than that of model SP when $\mu_z \leq 0.06$, as μ_z increases from 0.06, the worst-case performance of model RS gradually approaches that of model RO and outperforms that of model SP.

From RO_AVG, RS_AVG, and SP_AVG in Figure 3, we can observe that solutions obtained from model SP exhibit the best average performance. However, their advantage in the average performance over solutions of models RO and RS is relatively small, with a gap of less than 4%. This suggests that both models RO and RS are capable of producing solutions with good average performance. Additionally, Figure 3 shows that solutions obtained from model RS exhibit better average performance than that from model RO. By selecting appropriate values for parameter μ_z, the average performance of solutions obtained from model RS can be close to that of model SP.

Figure 3 also highlights that although parameter Γ is designed to control the level of robustness for the solutions of model RO, its effectiveness on the worst-case and average performances across the randomly generated scenarios is nonmonotonic and insignificant. Conversely, increasing parameter μ_z consistently and significantly improves both the worst-case and average performance of model RS, before reaching a plateau. Therefore, the cost target is effective in controlling the level of robustness for the solutions of model RS across the randomly generated scenarios.

Our above findings confirm the practical usefulness of the robust optimization model RO and the robust satisficing model RS in solving the CTSNDP under travel time uncertainty. First,

when compared to the stochastic programming model SP, both the RO and RS models are computationally more tractable. In addition, they do not necessitate complete information about the probability distribution of uncertain travel times, yet their average performances are comparable to that of the SP model. Second, compared to other models, model RO is capable of producing solutions that are more robust in worst-case performance and also have a good average performance. This makes it particularly useful for conservative decision makers who prioritize worst-case scenarios. However, as the price for robustness, decision makers who adopt model RO may incur higher costs in nominal scenarios. Third, the RS model is capable of generating solutions with better average performance than the RO model, and it can also closely approximate the average performance of the SP model. Thus, model RS can be particularly useful for decision makers who prioritize average performance but have limited distribution information. Furthermore, by setting its cost target properly, model RS is more effective in adjusting the trade-off between robustness against worst-case scenarios and costs in nominal scenarios. Although solutions obtained from model RS under some cost targets may exhibit a risk of higher worst-case costs than those obtained from model RO, their costs in nominal scenarios are typically lower. Therefore, model RS is useful for accommodating decision makers' different preferences towards robustness and its trade-offs.

6. Conclusions

This paper studies a robust continuous-time service network design problem (CTSNDP) under travel time uncertainty, aiming to design a transportation service network with reliable operational efficiency even under travel time deviation. Despite its importance, travel time uncertainty has seldom been explored in existing literature on the CTSNDP. This is because the time-indexed MILP models, which are commonly used to solve the CTSNDP with deterministic travel times, become impractical. To tackle this challenge, we derive a novel consolidation-indexed MILP model for the deterministic CTSNDP that eliminates the need for time indices. This enables us to derive a robust optimization model and a robust satisficing model to incorporate travel time uncertainty for the robust CTSNDP, based on polyhedral uncertainty sets. Both of these two models involve two stages of optimization, where the first stage involves routing and consolidation of shipments prior to actual travel times being realized, and the second stage involves departure schedules of vehicles after actual travel times are realized. To solve these two-stage optimization models to exact optimum, we derive several tractable reformulations for them, and based on these reformulations we develop two tailored column-and-constraint generation (C&CG) algorithms, respectively. To the best of our knowledge, this study stands out as the first to showcase the effectiveness of C&CG algorithms in solving two-stage robust satisficing models with a polyhedral uncertainty set. Furthermore, our

computational results provide compelling evidence regarding the tractability of the proposed models and the effectiveness of the developed algorithms. The robustness of the solutions obtained has been confirmed, and the trade-off between the robustness and its price has been highlighted.

This study has established a strong foundation for future research in several interesting and promising directions. First, there is great interest in enhancing our solution algorithms or developing new ones for the robust CTSNDP under travel time uncertainty. Our current C&CG algorithms rely on a general optimization solver to directly solve the MILP models of both subproblems and master problems. One possible enhancement is to develop tailored exact algorithms, such as branch-and-bound algorithms, to solve these MILP models more efficiently. This may require deriving tight and tractable relaxations of these MILP models. Moreover, our current C&CG algorithms, which are exact algorithms, have been shown to be efficient for instances with up to 50 commodities. It would be highly advantageous to improve or develop new algorithms, including heuristics, that can efficiently produce high-quality solutions for larger instances.

Second, as the first attempt at incorporating travel time uncertainty into robust service network design, we have developed robust optimization and robust satisficing models based on a polyhedral uncertainty set of the random variables that affect the actual travel times. Further exploration of alternative robust optimization approaches to tackle the robust CTSNDP under travel time uncertainty is an area of interest. For instance, the distributionally robust optimization approach (Goh and Sim 2010, Delage and Ye 2010, Mohajerin Esfahani and Kuhn 2018) could be utilized to identify robust service network designs that exhibit reliable performance for all possible probability distributions of uncertain travel times. These probability distributions could form an ambiguity set, satisfying certain conditions, such as moment conditions, that can be derived from historical travel time data. Therefore, solutions produced by the distributionally robust optimization approach are expected to perform better than those based on polyhedral uncertainty sets when historical data is available. To apply the distributionally robust optimization approach, optimization formulations with no time indices are still required, for which our newly proposed MILP formulation of the deterministic CTSNDP provides a solid base.

Thirdly, there is also significant potential for exploring additional applications of our newly proposed optimization models and solution algorithms. For instance, the MILP formulation of the deterministic CTSNDP, which we have introduced, exhibits flexibility by not necessitating time indices. In future studies, it could be further strengthened through the derivation of valid inequalities. This formulation could also be leveraged to develop novel exact and heuristic algorithms for the deterministic CTSNDP. Moreover, our robust optimization model and robust satisficing

models, accompanied by their C&CG algorithms, provide a solid foundation that can be extended and adapted to tackle travel time uncertainty in various other transportation problems. Examples include but are not limited to liner service network design problems (Wang and Meng 2012a,b, Lee et al. 2021) and aircraft routing and scheduling problems (Sohoni et al. 2011, Abdelghany and Abdelghany 2018, Yan and Kung 2018).

References

Abdelghany, A., Abdelghany, K., 2018. Airline network planning and scheduling. John Wiley & Sons.

Adulyasak, Y., Jaillet, P., 2016. Models and algorithms for stochastic and robust vehicle routing with deadlines. Transportation Science 50, 608–626.

Andersen, J., Christiansen, M., Crainic, T.G., Grønhaug, R., 2011. Branch and price for service network design with asset management constraints. Transportation Science 45, 33–49.

Andersen, J., Crainic, T.G., Christiansen, M., 2009a. Service network design with asset management: Formulations and comparative analyses. Transportation Research Part C: Emerging Technologies 17, 197–207.

Andersen, J., Crainic, T.G., Christiansen, M., 2009b. Service network design with management and coordination of multiple fleets. European Journal of Operational Research 193, 377–389.

Atamtürk, A., Zhang, M., 2007. Two-stage robust network flow and design under demand uncertainty. Operations Research 55, 662–673.

Bai, R., Wallace, S.W., Li, J., Chong, A.Y.L., 2014. Stochastic service network design with rerouting. Transportation Research Part B: Methodological 60, 50–65.

Ben-Tal, A., El Ghaoui, L., Nemirovski, A., 2009. Robust optimization. Princeton University Press.

Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., 2004. Adjustable robust solutions of uncertain linear programs. Mathematical Programming 99, 351–376.

Bertsimas, D., Brown, D.B., Caramanis, C., 2011. Theory and applications of robust optimization. SIAM Review 53, 464–501.

Bertsimas, D., Sim, M., 2004. The price of robustness. Operations Research 52, 35–53.

Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. Operations Research 65, 1303–1321.

Chassein, A., Goerigk, M., 2016. On the recoverable robust traveling salesman problem. Optimization Letters 10, 1479–1492.

Chen, B.Y., Li, Q., Lam, W.H., 2016. Finding the k reliable shortest paths under travel time uncertainty. Transportation Research Part B: Methodological 94, 189–203.

Chouman, M., Crainic, T.G., 2021. Freight railroad service network design, in: Network Design with Applications to Transportation and Logistics. Springer, pp. 383–426.

Crainic, T.G., Fu, X., Gendreau, M., Rei, W., Wallace, S.W., 2011. Progressive hedging-based metaheuristics for stochastic network design. Networks 58, 114–124.

Crainic, T.G., Hewitt, M., Toulouse, M., Vu, D.M., 2016. Service network design with resource constraints. Transportation Science 50, 1380–1393.

Crainic, T.G., Rousseau, J.M., 1986. Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. Transportation Research Part B: Methodological 20, 225–242.

Cui, Z., Ding, J., Long, D.Z., Zhang, L., 2023. Target-based resource pooling problem. Production and Operations Management 32, 1187–1204.

Delage, E., Ye, Y., 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. Operations Research 58, 595–612.

Demir, E., Burgholzer, W., Hrušovskỳ, M., Arıkan, E., Jammernegg, W., Van Woensel, T., 2016. A green intermodal service network design problem with travel time uncertainty. Transportation Research Part B: Methodological 93, 789–807.

Farvolden, J.M., Powell, W.B., 1994. Subgradient methods for the service network design problem. Transportation Science 28, 256–272.

Fischetti, M., Monaci, M., 2009. Light robustness, in: Robust and online large-scale optimization. Springer, pp. 61–84.

Ghamlouche, I., Crainic, T.G., Gendreau, M., 2003. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. Operations Research 51, 655–667.

Goh, J., Sim, M., 2010. Distributionally robust optimization and its tractable approximations. Operations Research 58, 902–917.

Hellsten, E., Koza, D.F., Contreras, I., Cordeau, J.F., Pisinger, D., 2021. The transit time constrained fixed charge multi-commodity network design problem. Computers & Operations Research 136, 105511.

Hewitt, M., 2019. Enhanced dynamic discretization discovery for the continuous time load plan design problem. Transportation Science 53, 1731–1750.

Hewitt, M., Lehuédé, F., 2023. New formulations for the scheduled service network design problem. Transportation Research Part B: Methodological 172, 117–133.

Hoff, A., Lium, A.G., Løkketangen, A., Crainic, T.G., 2010. A metaheuristic for stochastic service network design. Journal of Heuristics 16, 653–679.

Hrušovskỳ, M., Demir, E., Jammernegg, W., Van Woensel, T., 2018. Hybrid simulation and optimization approach for green intermodal transportation problem with travel time uncertainty. Flexible Services and Manufacturing Journal 30, 486–516.

Hu, C., Lu, J., Liu, X., Zhang, G., 2018. Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. Computers & Operations Research 94, 139–153.

Koster, A.M., Kutschka, M., Raack, C., 2013. Robust network design: Formulations, valid inequalities, and computations. Networks 61, 128–149.

Lanza, G., Crainic, T.G., Rei, W., Ricciardi, N., 2021. Scheduled service network design with quality targets and stochastic travel times. European Journal of Operational Research 288, 30–46.

Layeb, S.B., Jaoua, A., Jbira, A., Makhlouf, Y., 2018. A simulation-optimization approach for scheduling in stochastic freight transportation. Computers & Industrial Engineering 126, 99–110.

Lee, C.Y., Shu, S., Xu, Z., 2021. Optimal global liner service procurement by utilizing liner service schedules. Production and Operations Management 30, 703–714.

Liang, J., Wu, J., Qu, Y., Yin, H., Qu, X., Gao, Z., 2019. Robust bus bridging service design under rail transit system disruptions. Transportation Research Part E: Logistics and Transportation Review 132, 97–116.

Lium, A.G., Crainic, T.G., Wallace, S.W., 2009. A study of demand stochasticity in service network design. Transportation Science 43, 144–157.

Long, D.Z., Sim, M., Zhou, M., 2023. Robust satisficing. Operations Research 71, 61–82.

Marshall, L., Boland, N., Savelsbergh, M., Hewitt, M., 2021. Interval-based dynamic discretization discovery for solving the continuous-time service network design problem. Transportation Science 55, 29–51.

Mohajerin Esfahani, P., Kuhn, D., 2018. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming 171, 115–166.

Pedersen, M.B., Crainic, T.G., Madsen, O.B., 2009. Models and tabu search metaheuristics for service network design with asset-balance requirements. Transportation Science 43, 158–177.

Remli, N., Amrouss, A., El Hallaoui, I., Rekik, M., 2019. A robust optimization approach for the winner determination problem with uncertainty on shipment volumes and carriers' capacity. Transportation Research Part B: Methodological 123, 127–148.

Sarayloo, F., Crainic, T.G., Rei, W., 2021a. A learning-based matheuristic for stochastic multicommodity network design. INFORMS Journal on Computing 33, 643–656.

Sarayloo, F., Crainic, T.G., Rei, W., 2021b. A reduced cost-based restriction and refinement matheuristic for stochastic network design problem. Journal of Heuristics 27, 325–351.

Scherr, Y.O., Saavedra, B.A.N., Hewitt, M., Mattfeld, D.C., 2019. Service network design with mixed autonomous fleets. Transportation Research Part E: Logistics and Transportation Review 124, 40–55.

Shu, S., Xu, Z., Baldacci, R., 2023. Incorporating holding costs in continuous-time service network design: New model, relaxation, and exact algorithm. Available at https://optimization-online.org/2021/10/8616/ .

Simchi-Levi, D., Wang, H., Wei, Y., 2019. Constraint generation for two-stage robust network flow problems. INFORMS Journal on Optimization 1, 49–70.

Sohoni, M., Lee, Y.C., Klabjan, D., 2011. Robust airline scheduling under block-time uncertainty. Transportation Science 45, 451–464.

Thiele, A., Terry, T., Epelman, M., 2009. Robust linear optimization with recourse. Rapport Technique , 4–37.

Wang, S., Meng, Q., 2012a. Liner ship route schedule design with sea contingency time and port time uncertainty. Transportation Research Part B: Methodological 46, 615–633.

Wang, S., Meng, Q., 2012b. Robust schedule design for liner shipping services. Transportation Research Part E: Logistics and Transportation Review 48, 1093–1106.

Wang, Z., Qi, M., 2020. Robust service network design under demand uncertainty. Transportation Science 54, 676–689.

Wieberneit, N., 2008. Service network design for freight transportation: a review. OR spectrum 30, 77–112.

Yan, C., Kung, J., 2018. Robust aircraft routing. Transportation Science 52, 118–133.

Yanıkoğlu, İ., Gorissen, B.L., den Hertog, D., 2019. A survey of adjustable robust optimization. European Journal of Operational Research 277, 799–813.

Yao, B., Hu, P., Lu, X., Gao, J., Zhang, M., 2014. Transit network design based on travel time reliability. Transportation Research Part C: Emerging Technologies 43, 233–248.

Zeng, B., Zhao, L., 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. Operations Research Letters 41, 457–461.

Zhang, Y., Zhang, Z., Lim, A., Sim, M., 2021. Robust data-driven vehicle routing with time windows. Operations Research 69, 469–485.

Zhao, Y., Xue, Q., Cao, Z., Zhang, X., 2018. A two-stage chance constrained approach with application to stochastic intermodal service network design problems. Journal of Advanced Transportation 2018, 1–18.

Zhou, M., Sim, M., Lam, S.W., 2022. Advance admission scheduling via resource satisficing. Production and Operations Management 31, 4002–4020.

Zhu, E., Crainic, T.G., Gendreau, M., 2014. Scheduled service network design for freight rail transportation. Operations Research 62, 383–400.

Glossaries of Notation, Proofs of Statements, Acceleration Strategies for C&CG Algorithms, and Details on Instance Generation for Computational Experiments

Appendix A: Glossaries of Notation

We summarize the notation used for problem description and solution representation in Table A.1, and for various optimization models in Table A.2.

Table A.1 Glossary of notation used: Problem description and solution representation

Notation	Meaning
$\mathcal{D}$	(flat) network $\mathcal{D}=(\mathcal{N},\mathcal{A})$
$\mathcal{N}$	node set of network $\mathcal{D}$
$\mathcal{A}$	arc set of network $\mathcal{D}$
$\mathcal{K}$	set of commodities
o^k	origin of commodity $k\in\mathcal{K}$
d^k	destination of commodity $k\in\mathcal{K}$
q^k	demand of commodity $k\in\mathcal{K}$
τ_{ij}	travel time of arc $(i,j)\in\mathcal{A}$
c_{ij}^k	per-unit-of-flow cost of arc $(i,j)\in\mathcal{A}$ and commodity $k\in\mathcal{K}$
f_{ij}	fixed cost of arc $(i,j)\in\mathcal{A}$
u_{ij}	capacity of arc $(i,j)\in\mathcal{A}$
e^k	earliest available time of commodity $k\in\mathcal{K}$
l^k	latest arrival time of commodity $k\in\mathcal{K}$
h^k	per-unit-of-demand-and-time (in-storage holding) cost of commodity $k\in\mathcal{K}$ at a terminal
$\alpha^{(r)}$	the r-th consolidation traveling arc duplicated from arc α of the flat network
$\mathcal{P}=\{P^k\}_{k\in\mathcal{K}}$	a routing plan with P^k representing a path for commodity $k\in\mathcal{K}$
$\mathcal{C}=\{C_r^\alpha\}_{\alpha\in\mathcal{A},r\in\{1,2,...,\lvert\mathcal{K}\rvert\}}$	a consolidation plan with each C_r^α being a subset of commodities and r denoting the consolidation index
$\mathcal{T}$	a departure schedule
$f(\mathcal{P},\mathcal{C})$	the total fixed cost and flow cost for a solution $(\mathcal{P},\mathcal{C},\mathcal{T})$
$h(\mathcal{P},\mathcal{T})$	the total holding cost for solution $(\mathcal{P},\mathcal{C},\mathcal{T})$
$g(\mathcal{P},\mathcal{T})$	the total delay penalty for solution $(\mathcal{P},\mathcal{C},\mathcal{T})$
$\boldsymbol{\delta}$	a vector of random variables $\delta_{\alpha r}$ for $\alpha\in\mathcal{A}$ and $r\in\{1,2,...,\lvert\mathcal{K}\rvert\}$
$\tilde{\tau}_{ij}$	uncertain travel time of arc $(i,j)\in\mathcal{A}$
$\tilde{\boldsymbol{\tau}}$	a vector of uncertain travel times $\tilde{\tau}_{ij}$
$\bar{\tau}_{ij}$	nominal value of $\tilde{\tau}_{ij}$
$\hat{\tau}_{ij}$	maximum deviation of $\tilde{\tau}_{ij}$ with respect to the nominal value $\bar{\tau}_{ij}$
$\mathcal{Z}$	a prescribed target of the total two-stage cost
$\mathbb{U}$	the support of the vector $\boldsymbol{\delta}$
$\mathbb{U}(\Gamma)$	budgeted uncertainty set of vector $\boldsymbol{\delta}$ with Γ denoting the budget of uncertainty
$\mathbb{D}$	the domain of all feasible solutions $(\mathcal{P},\mathcal{C},\mathcal{T})$
$\mathbb{F}$	the domain of all nominal timely-implementable first-stage solutions
$\mathbb{T}(\mathcal{P},\mathcal{C},\tilde{\boldsymbol{\tau}})$	the domain of departure schedule $\mathcal{T}$ with respect to solution $(\mathcal{P},\mathcal{C})$

Table A.2 Glossary of notation used: Models

Notation	Meaning
$\boldsymbol{x}$	routing decision variables
$\boldsymbol{z}$	consolidation decision variables
$\boldsymbol{y}$	service decision variables
$\boldsymbol{v}$	decision variables on departure times
$\mathcal{Z}$, $\mathcal{Q}$	domains of specific variables
DO	consolidation-indexed formulation for the deterministic CTSNDP
RO	robust optimization model for the robust CTSNDP
RS	robust satisficing model for the robust CTSNDP
RP$(\boldsymbol{x},\boldsymbol{z})$	the max-min model defined by (3.29)–(3.40) for calculating the worst-case second-stage cost
LP$(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}})$	the inner minimization problem of model RP$(\boldsymbol{x},\boldsymbol{z})$
DLP$(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}})$	the dual linear problem of model LP$(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}})$
ROMILP	noncompact MILP defined in (4.1)-(4.4), a reformulation of model RO
RSMILP	noncompact MILP reformulation of model RS, containing constraints (4.27)-(4.28)
FO$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$	the model defined in (4.29) for calculating the worst-case normalized cost deviation of solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$
$F_1(\boldsymbol{x},\boldsymbol{y})$	the first-stage cost for solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$
$F_{RP}(\boldsymbol{x},\boldsymbol{z})$	the worst-case second-stage cost for solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$
$F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}})$	the optimal objective value of model LP$(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}})$, indicating the minimum cost for solution $(\boldsymbol{x},\boldsymbol{z})$ under $\tilde{\boldsymbol{\tau}}$
$\rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$	the optimal objective value of model FO$(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$
$G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\hat{\rho})$	the optimal objective value of the model defined in (4.30)

Appendix B: Proof of Statements

B.1. Proof of Lemma 4.1

For any given $(\boldsymbol{x},\boldsymbol{z})$ that satisfies constraints (3.20)–(3.28) of model RO, it corresponds to a nominal timely-implementable flat solution $(\mathcal{P},\mathcal{C})$. Consider any $\boldsymbol{\delta}\in\mathbb{U}(\Gamma)$ with the corresponding realized travel time $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$. For such $(\mathcal{P},\mathcal{C})$ and $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$, we first show as follows that there exists a departure schedule $\mathcal{T}$ such that constraints (2.1)–(2.4) are satisfied, from which we can then obtain a feasible solution to model LP$(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$.

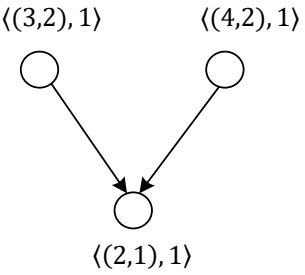
For the nominal timely-implementable flat solution $(\mathcal{P},\mathcal{C})$, consider each commodity $k\in\mathcal{K}$ and its flat path P^k in $\mathcal{P}$ with an arc sequence denoted by $(a^k_1,...,a^k_{m^k})$. For each $n\in\{1,2,\ldots,m^k\}$, there must exist a consolidation $C^{a^k_n}_{r^k_n}\in\mathcal{C}$ for arc a^k_n with $r^k_n\in\{1,2,\cdots,|\mathcal{K}|\}$ such that $k\in C^{a^k_n}_{r^k_n}$. We can now construct a network $\mathcal{G}_{\mathcal{C}}=\{\mathcal{N}_{\mathcal{C}},\mathcal{A}_{\mathcal{C}}\}$ where each non-empty consolidation $C^{\alpha}_{r}\in\mathcal{C}$ corresponds to a node, denoted by $\langle\alpha,r\rangle$, in the node set $\mathcal{N}_{\mathcal{C}}$, and each pair of consolidations $C^{a^k_n}_{r^k_n}$ and $C^{a^k_{n+1}}_{r^k_{n+1}}$ for $k\in\mathcal{K}$ and $n\in\{1,...,m^k-1\}$ corresponds to an arc $(\langle a^k_n,r^k_n\rangle,\langle a^k_{n+1},r^k_{n+1}\rangle)$ in the arc set $\mathcal{A}_{\mathcal{C}}$. See Figure 4 for an example of such a network $\mathcal{G}_{\mathcal{C}}$.

Figure 4 An example of network $\mathcal{G}_\mathcal{C}$ constructed from a given nominal timely-implementable flat solution $(\mathcal{P},\mathcal{C})$.

(a) Routing plan $\mathcal{P}$, consolidation plan $\mathcal{C}$, and consolidations along the path P^k for each commodity $k \in \mathcal{K}$

$\mathcal{P}$	$\mathcal{C}$	$\{C_{r_n^k}^{a_n^k}\}$
P^{k_1}: (2,1)	$C_1^{(2,1)}$: $\{k_1, k_2, k_3\}$	k_1: $C_1^{(2,1)}$
P^{k_2}: (4,2) → (2,1)	$C_1^{(3,2)}$: $\{k_3\}$	k_2: $C_1^{(4,2)} \to C_1^{(2,1)}$
P^{k_3}: (3,2) → (2,1)	$C_1^{(4,2)}$: $\{k_2,\}$	k_3: $C_1^{(3,2)} \to C_1^{(2,1)}$

(b) The resulting network $\mathcal{G}_\mathcal{C}$



Since the flat solution $(\mathcal{P},\mathcal{C})$ is a nominal timely-implementable first-stage solution, there exists a departure schedule $\mathcal{T}$ which satisfies (2.1)–(2.4) with nominal travel times $\overline{\boldsymbol{\tau}}$. According to $\mathcal{T}$, for each consolidation $C_r^\alpha \in \mathcal{C}$ of arc $\alpha = (\nu, \nu') \in \mathcal{A}$ we can obtain its corresponding departure time from node ν, which is denoted by $t_{\alpha,r}$. For each pair of consolidations $C_{r_n^k}^{a_n^k}$ and $C_{r_{n+1}^k}^{a_{n+1}^k}$ with $k \in \mathcal{K}$ and $n \in \{1, ..., m^k - 1\}$, the departure time of $C_{r_n^k}^{a_n^k}$ from node ν_n^k plus the nominal value $\overline{\tau}_{a_n^k}$ of travel time of arc a_n^k must be less than or equal to the departure time of $C_{r_{n+1}^k}^{a_{n+1}^k}$ from node ν_{n+1}^k. Thus, by the definition of $\mathcal{G}_\mathcal{C} = \{\mathcal{N}_\mathcal{C}, \mathcal{A}_\mathcal{C}\}$, we obtain that

$$t_{\alpha,r} + \overline{\tau}_\alpha \le t_{\alpha',r'}, \ \forall\ (\langle \alpha, r\rangle, \langle \alpha', r'\rangle) \in \mathcal{A}_\mathcal{C}.$$

This, together with $\overline{\tau}_\alpha > 0$ for all $\alpha \in \mathcal{A}$, implies that $\mathcal{G}_\mathcal{C}$ must be an acyclic network, and thus has a topological ordering of nodes in $\mathcal{N}_\mathcal{C}$, denoted by $(\langle \alpha_1, r_1\rangle, \langle \alpha_2, r_2\rangle, \ldots, \langle \alpha_{|\mathcal{N}_\mathcal{G}|}, r_{|\mathcal{N}_\mathcal{G}|}\rangle)$.

Next, consider each possible realized travel time $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$ with any $\boldsymbol{\delta} \in \mathbb{U}(\Gamma)$. For $n = 1, 2, \ldots, |\mathcal{N}_\mathcal{G}|$, we can set the departure time of consolidation $C_{r_n}^{\alpha_n}$, denoted by $\hat{t}_{\alpha_n, r_n}$, iteratively as follows:

$$\begin{aligned}
&\hat{t}_{\alpha_1, r_1} = \max_{k \in \mathcal{K}} e^k, \\
&\hat{t}_{\alpha_n, r_n} = \hat{t}_{\alpha_1, r_1} + \max_{(i,j) \in \mathcal{A}} \{\overline{\tau}_{ij} + \hat{\tau}_{ij}\} \quad \text{for } n = 2, 3, \ldots, |\mathcal{N}_\mathcal{C}|.
\end{aligned}$$

Essentially, the departure schedule determined above follows a reactive policy in which for each $(i,j) \in \mathcal{A}$, every consolidation on arc (i,j) departs from node i as soon as all its commodities arrive at i. Thus, it can be seen that for each commodity $k \in \mathcal{K}$,

$$\begin{aligned}
&\hat{t}_{\alpha_1^k, r_1^k} \ge \hat{t}_{\alpha_1, r_1} = \max_{k \in \mathcal{K}} e^k \ge e^k, \\
&\hat{t}_{\alpha_{n+1}^k, r_{n+1}^k} \ge \hat{t}_{\alpha_n^k, r_n^k} + \max_{(i,j) \in \mathcal{A}} \{\overline{\tau}_{ij} + \hat{\tau}_{ij}\} \ge \hat{t}_{\alpha_n^k, r_n^k} + \tilde{\tau}_{\alpha_n^k} \quad \text{for } n = 1, \ldots, m^k - 2.
\end{aligned}$$

Thus, by setting the departure time of commodity k for node ν_n^k to be equal to $\hat{t}_{\alpha_n^k, r_n^k}$, for $n = 1, 2, \ldots, m_k - 1$ and $k \in \mathcal{K}$, we obtain a plan $\hat{\mathcal{T}}$ which satisfies the constraints (2.1), (2.2) and (2.4) under the travel time $\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})$. From such a departure schedule $\hat{\mathcal{T}}$, we can obtain the values of variables v_{ij}^k, b_{ijr}, w_i^k, and s^k according to their definitions, which form a feasible solution to model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))$. Hence, Lemma 4.1 is proved. □

B.2. Proof of Proposition 4.1

Proof. For any given $(\boldsymbol{x}, \boldsymbol{z})$, consider any optimal solution $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\psi}^*, \boldsymbol{\eta}^*, \boldsymbol{\theta}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*, \tilde{\boldsymbol{\tau}}^*, \boldsymbol{\delta}^*)$ of the nonlinear optimization model defined in (4.14)–(4.18). By fixing $(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\lambda}) = (\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\psi}^*, \boldsymbol{\eta}^*, \boldsymbol{\theta}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*)$, the nonlinear optimization model defined in (4.14)–(4.18) reduces to the following nonlinear model on $\boldsymbol{\delta}$, denoted as model $\mathbf{R}_1$.

$$
\begin{aligned}
[\mathbf{R}_1] \quad \max \quad & \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big\{ \hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*}) + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \cdot \delta_{jir} \Big\} \\
s.t. \quad & -1 \le \delta_{ijr} \le 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\}, \\
& \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} |\delta_{ijr}| \le \Gamma.
\end{aligned}
$$

It can be seen that $\boldsymbol{\delta}^*$ must be an optimal solution to model $\mathbf{R}_1$. Moreover, for any optimal solution $\hat{\boldsymbol{\delta}}$ to model $\mathbf{R}_1$, $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\psi}^*, \boldsymbol{\eta}^*, \boldsymbol{\theta}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*, \tilde{\boldsymbol{\tau}}(\hat{\boldsymbol{\delta}}), \hat{\boldsymbol{\delta}})$ forms a feasible solution to the nonlinear optimization model defined in (4.14)–(4.18), and it has the same objective value as that of $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\psi}^*, \boldsymbol{\eta}^*, \boldsymbol{\theta}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*, \tilde{\boldsymbol{\tau}}^*, \boldsymbol{\delta}^*)$. Thus, $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\psi}^*, \boldsymbol{\eta}^*, \boldsymbol{\theta}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*, \tilde{\boldsymbol{\tau}}(\hat{\boldsymbol{\delta}}), \hat{\boldsymbol{\delta}})$ is also an optimal solution to the nonlinear optimization model defined in (4.14)–(4.18).

Consider any optimal solution $\hat{\boldsymbol{\delta}}$ to model $\mathbf{R}_1$. Due to the optimality of $\hat{\boldsymbol{\delta}}$, it can be seen that for any $(j,i) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$, if $\hat{\delta}_{jir} > 0$, then $\hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*} + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \ge 0$, and that if $\hat{\delta}_{jir} < 0$, then $\hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*} + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \le 0$. This is because otherwise, $\hat{\boldsymbol{\delta}}$ cannot be an optimal solution to model $\mathbf{S}_1$, as we can increase its objective value by changing the sign of each $\hat{\delta}_{jir}$ with $\hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*}) + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \cdot \hat{\delta}_{jir} < 0$ to its opposite. Thus, we obtain that $\hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*}) + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \cdot \hat{\delta}_{jir} \ge 0$ for all $(j,i) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. Accordingly, model $\mathbf{R}_1$ is equivalent to the following maximization LP, denoted as model $\mathbf{R}_2$:

$$
\begin{aligned}
[\mathbf{R}_2] \quad \max \quad & \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big\{ \Big| \hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*} + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \Big| \cdot \delta^+_{jir} \Big\} \\
s.t. \quad & \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \delta^+_{ijr} \le \Gamma, \\
& 0 \le \delta^+_{ijr} \le 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\}.
\end{aligned}
$$

From any optimal solution $\boldsymbol{\delta}^+$ to model $\mathbf{R}_2$, we can derive an optimal solution to model $\mathbf{R}_1$ by setting $\delta_{jir} = \delta^+_{jir}$ if $\hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*} + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) \ge 0$, and setting $\delta_{jir} = -\delta^+_{jir}$ if $\hat{\tau}_{jir} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir} (\beta_i^{k^*} - \lambda_i^{k^*} + \sum_{k\in\mathcal{K}_i^d} z^k_{jir} (\psi^{k^*} - \lambda_i^{k^*}) \Big) < 0$, for each $(j,i) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$, so that their objective values are the same.

For model $\mathbf{R}_2$, its constraint matrix associated with $\sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \delta^+_{ijr} \le \Gamma$ and $\delta^+_{ijr} \le 1$ for all $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$ is totally unimodular, as it contains two entries of 1 in each column. This implies that with an integral Γ, the feasible solution region of model $\mathbf{R}_2$ is an integral polytope. Thus, there exists an integral optimal solution to model $\mathbf{R}_2$ with $\delta^+_{ijr} \in \{0,1\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. This implies that

there exists an optimal solution $\boldsymbol{\delta}$ to model $\mathbf{R}_1$ with $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. Therefore, there exists an optimal solution to the nonlinear optimization model defined in (4.14)–(4.18) that satisfies $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. Hence, Proposition 4.1 is proved. □

B.3. Proof of Proposition 4.2

Proof. We first note that the max-min model RP($\boldsymbol{x}, \boldsymbol{z}$) defined by (3.29)–(3.40) for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ can be reformulated to the nonlinear optimization model defined in (4.14)–(4.18). We then prove Proposition 4.2 by showing that the nonlinear optimization model defined in (4.14)–(4.18) can be linearized to the MILP model defined by (4.19)–(4.24).

By Proposition 4.1, constraints (4.17) can be replaced with $\delta_{ijr} \in \{-1, 0, 1\}$ for all $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. By (4.16) we have that $\tilde{\tau}_{ijr} \in \{\overline{\tau}_{ijr} - \hat{\tau}_{ijr}, \overline{\tau}_{ijr}, \overline{\tau}_{ijr} + \hat{\tau}_{ijr}\}$, which, together with $\overline{\tau}_{ijr} \in \mathbb{N}_{>0}$, $\hat{\tau}_{ijr} \in \mathbb{N}_0$ and $\overline{\tau}_{ijr} > \hat{\tau}_{ijr}$, implies that $\tilde{\tau}_{ijr} \in \mathbb{N}_{>0}$. Moreover, we introduce a new variable φ_{jir} to represent each nonlinear term $\Big(\sum_{k \in \mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k \in \mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big) \cdot \tilde{\tau}_{jir}$. We then replace each integer variable δ_{jir} with three new binary variables $\zeta_{jir,-1}$, $\zeta_{jir,0}$ and $\zeta_{jir,1}$, which are used to indicate whether δ_{ijr} equals -1, 0 and 1, respectively. Let $\tilde{\tau}_{ijr,-1} = \overline{\tau}_{ijr} - \hat{\tau}_{ijr}$, $\tilde{\tau}_{ijr,0} = \overline{\tau}_{ijr}$ and $\tilde{\tau}_{ijr,1} = \overline{\tau}_{ijr} + \hat{\tau}_{ijr}$. Accordingly, the following linear constraints can be derived for the newly introduced variables, where M_2 is a sufficiently large constant.

$$\zeta_{ijr,-1} + \zeta_{ijr,0} + \zeta_{ijr,1} = 1, \quad \forall\ (i, j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{B.1}$$

$$\Big(\sum_{k \in \mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k \in \mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big)\tilde{\tau}_{jir,\ell} - M_2(1 - \zeta_{jir,\ell}) \le \varphi_{jir}$$
$$\le \Big(\sum_{k \in \mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k \in \mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big)\tilde{\tau}_{jir,\ell} + M_2(1 - \zeta_{jir,\ell}),$$
$$\forall\ (j, i) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \ell = \{-1, 0, 1\}, \tag{B.2}$$

$$\zeta_{ijr,\ell} \in \{0, 1\}, \quad \forall\ (i, j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \ell = \{-1, 0, 1\}. \tag{B.3}$$

Here, constraints (B.1) ensure that exactly one of the three variables $\zeta_{jir,-1}$, $\zeta_{jir,0}$ and $\zeta_{jir,1}$ equals 1, constraints (B.2) ensure that each variable φ_{jir} equals $\Big(\sum_{k \in \mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k \in \mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big) \cdot \tilde{\tau}_{jir}$, and constraints (B.3) define the domain of the variables $\zeta_{ijr,-1}$, and $\zeta_{ijr,0}$, and $\zeta_{ijr,1}$. As a result, constraints (4.18) can be replaced with the following linear constraint:

$$\sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} (\zeta_{ijr,-1} + \zeta_{ijr,1}) \le \Gamma. \tag{B.4}$$

Therefore, the nonlinear optimization model defined by (4.14)–(4.18) for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ can be further reformulated to the following maximization MILP model:

$$\begin{aligned} F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \max \quad & \sum_{(j,i) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \varphi_{jir} - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (M_1 x^k_{ij}) \cdot \eta^k_{ij} \\ & + \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1(z^k_{ijr} - 1)] \cdot (\theta^k_{ijr} + \xi^k_{ijr}) \\ & + \sum_{k \in \mathcal{K}} e^k \cdot (\gamma^k - \lambda^k_{o^k}) + \sum_{k \in \mathcal{K}} l^k \cdot (\lambda^k_{d^k} - \psi^k) \\ \text{s.t.} \quad & (4.6) - (4.13), (B.1) - (B.3) \text{ and } (B.4). \end{aligned}$$

Hence, Proposition 4.2 is proved. □

B.4. Proof of Theorem 4.1

At each iteration of Algorithm 1, UB and LB are updated by solving the corresponding master problem and subproblem, while a new worst-case scenario $\boldsymbol{\delta}$ in $\mathbb{U}(\Gamma)$ is obtained and added to the scenario subset Λ. Algorithm 1 stops when $UB = LB$. As model ROMILP(Λ) is a relaxation of the reformulation ROMILP of model RO, the value of LB, which equals the optimal objective value of model ROMILP(Λ), is a valid lower bound on the optimal objective value of model RO. As UB is the worst-case total cost of the first-stage solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ obtained from the master problem, it provides a valid upper bound on the optimal objective value of model RO. Thus, when $UB = LB$, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ must be an optimal solution to model RO. This implies that when Algorithm 1 terminates with $UB = LB$, it returns an optimal solution to model RO.

We next show as follows that Algorithm 1 terminates with $UB = LB$ in a finite number of iterations. To show this, we note that, at each iteration n, if the worst-case scenario $\boldsymbol{\delta}^{(n)}$ identified in Step 3 of Algorithm 1 is not in the current scenario subset Λ, it will be added to Λ. According to Proposition 4.1, $\boldsymbol{\delta}^{(n)}$ satisfies that $\boldsymbol{\delta}^n_{ijr} \in \{-1, 0, 1\}$ for all $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, \cdots, |\mathcal{K}|\}$, and has a finite number of possible values. Therefore, in a finite number of iterations, $\boldsymbol{\delta}^{(n)}$ identified in Step 3 of Algorithm 1 must be included in the current scenario subset Λ. In such a situation, both LB and UB must be equal to the optimal objective value of the current master problem, implying that $LB = UB$ and $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ forms an optimal solution to model RO. This completes the proof of Theorem 4.1. □

B.5. Proof of Lemma 4.2

Recall that we slightly abuse the notation to define that $\sigma/\|\mathbf{0}\|_1 = 0$ for $\sigma = 0$, $\sigma/\|\mathbf{0}\|_1 = +\infty$ for $\sigma > 0$, and $\sigma/\|\mathbf{0}\|_1 = -\infty$ for $\sigma < 0$. Consider any $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in \mathcal{X}$ and any given $\hat{\rho}$.

First, if $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) > 0$, then according to (4.30), we have that

$$\max_{\boldsymbol{\delta} \in \mathbb{U}} \ \{F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \hat{\rho}\|\boldsymbol{\delta}\|_1\} > 0,$$

which implies that there exists a $\boldsymbol{\delta}^* \in \mathbb{U}$ with

$$F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}^*)) - \mathcal{Z} - \hat{\rho}\|\boldsymbol{\delta}^*\|_1 > 0.$$

Thus, we obtain that

$$\rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \max_{\boldsymbol{\delta} \in \mathbb{U}} \ \frac{F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}}{\|\boldsymbol{\delta}\|_1} \geq \frac{F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}^*)) - \mathcal{Z}}{\|\boldsymbol{\delta}^*\|_1} > \hat{\rho}$$

Second, if $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \hat{\rho}) < 0$, then we have that

$$\max_{\boldsymbol{\delta} \in \mathbb{U}} \ \{F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \hat{\rho}\|\boldsymbol{\delta}\|_1\} < 0,$$

which implies that

$$F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \hat{\rho}\|\boldsymbol{\delta}\|_1 < 0, \quad \forall \boldsymbol{\delta} \in \mathbb{U}.$$

Thus,

$$\frac{F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))-\mathcal{Z}}{\|\boldsymbol{\delta}\|_1}<\hat{\rho},\quad \forall \boldsymbol{\delta}\in\mathbb{U}.$$

Therefore, we obtain that

$$\rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})=\max_{\boldsymbol{\delta}\in\mathbb{U}}\ \frac{F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))-\mathcal{Z}}{\|\boldsymbol{\delta}\|_1}<\hat{\rho}.$$

Third, if $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\hat{\rho})=0$, then we obtain that

$$\max_{\boldsymbol{\delta}\in\mathbb{U}}\ \{F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))-\mathcal{Z}-\hat{\rho}\|\boldsymbol{\delta}\|_1\}=0,$$

which implies that there exists a $\boldsymbol{\delta}^*\in\mathbb{U}$ such that

$$\begin{aligned}&F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}^*))-\mathcal{Z}-\hat{\rho}\|\boldsymbol{\delta}^*\|_1=0,\ \text{and}\\ &F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))-\mathcal{Z}-\hat{\rho}\|\boldsymbol{\delta}\|_1\le 0,\quad \forall \boldsymbol{\delta}\in\mathbb{U}\setminus\{\boldsymbol{\delta}^*\}.\end{aligned}$$

Thus, we have that

$$\begin{aligned}&\frac{F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}^*))-\mathcal{Z}}{\|\boldsymbol{\delta}^*\|_1}=\hat{\rho},\ \text{and}\\ &\frac{F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))-\mathcal{Z}}{\|\boldsymbol{\delta}\|_1}\le\hat{\rho},\quad \forall \boldsymbol{\delta}\in\mathbb{U}\setminus\{\boldsymbol{\delta}^*\},\end{aligned}$$

which implies that

$$\hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})=\max_{\boldsymbol{\delta}\in\mathbb{U}}\ \frac{F_1(\boldsymbol{x},\boldsymbol{y})+F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}))-\mathcal{Z}}{\|\boldsymbol{\delta}\|_1}=\hat{\rho}.$$

Hence, Lemma 4.2 is proved. □

B.6. Proof of Proposition 4.3

Proof. Our proof of Proposition 4.3 follows an argument similar to that of Proposition 4.1. For any given $(\boldsymbol{x},\boldsymbol{z})$ and $\hat{\rho}$, consider any optimal solution $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*,\boldsymbol{\psi}^*,\boldsymbol{\eta}^*,\boldsymbol{\theta}^*,\boldsymbol{\xi}^*,\boldsymbol{\lambda}^*,\tilde{\boldsymbol{\tau}}^*,\boldsymbol{\delta}^*)$ of the optimization model defined in (4.31)–(4.34). By fixing $(\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\psi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\xi},\boldsymbol{\lambda})=(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*,\boldsymbol{\psi}^*,\boldsymbol{\eta}^*,\boldsymbol{\theta}^*,\boldsymbol{\xi}^*,\boldsymbol{\lambda}^*)$, the nonlinear optimization model defined in (4.31)–(4.34) reduces to the following nonlinear model on $\boldsymbol{\delta}$, denoted as model $\mathbf{S}_1$.

$$\begin{aligned}[\mathbf{S}_1]\ \max\quad &\sum_{(j,i)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}\Big\{\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i}z^k_{jir}(\beta_i^{k^*}-\lambda_i^{k^*})+\sum_{k\in\mathcal{K}_i^d}z^k_{jir}(\psi^{k^*}-\lambda_i^{k^*})\Big)\cdot\delta_{jir}-\hat{\rho}\cdot|\delta_{jir}|\Big\}\\ s.t.\quad &-1\le\delta_{ijr}\le 1,\quad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}\end{aligned}$$

It can be seen that $\boldsymbol{\delta}^*$ is an optimal solution of model $\mathbf{S}_1$. For any optimal solution $\hat{\boldsymbol{\delta}}$ of model $\mathbf{S}_1$, $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*,\boldsymbol{\psi}^*,\boldsymbol{\eta}^*,\boldsymbol{\theta}^*,\boldsymbol{\xi}^*,\boldsymbol{\lambda}^*,\tilde{\boldsymbol{\tau}}(\hat{\boldsymbol{\delta}}),\hat{\boldsymbol{\delta}})$ forms a feasible solution of the optimization model defined in (4.31)–(4.34), and it has the same objective value as that of $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*,\boldsymbol{\psi}^*,\boldsymbol{\eta}^*,\boldsymbol{\theta}^*,\boldsymbol{\xi}^*,\boldsymbol{\lambda}^*,\tilde{\boldsymbol{\tau}}^*,\boldsymbol{\delta}^*)$. Thus, $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*,\boldsymbol{\psi}^*,\boldsymbol{\eta}^*,\boldsymbol{\theta}^*,\boldsymbol{\xi}^*,\boldsymbol{\lambda}^*,\tilde{\boldsymbol{\tau}}(\hat{\boldsymbol{\delta}}),\hat{\boldsymbol{\delta}})$ is also an optimal solution to the optimization model defined in (4.31)–(4.34).

Consider any optimal solution $\hat{\boldsymbol{\delta}}$ to model $\mathbf{S}_1$. Due to the optimality of $\hat{\boldsymbol{\delta}}$, it can be seen that for any $(j,i)\in\mathcal{A}$ and $r\in\{1,2,...,|\mathcal{K}|\}$, if $\hat{\delta}_{jir}>0$, then $\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i}z^k_{jir}(\beta_i^{k^*}-\lambda_i^{k^*}+\sum_{k\in\mathcal{K}_i^d}z^k_{jir}(\psi^{k^*}-\lambda_i^{k^*})\Big)\ge 0$, and

that if $\hat{\delta}_{jir} < 0$, then $\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^{k*}_i - \lambda^{k*}_i + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^{k*} - \lambda^{k*}_i)\Big) \le 0$. This is because otherwise, $\hat{\boldsymbol{\delta}}$ cannot be an optimal solution to model $\mathbf{S}_1$, as we can increase its objective value by changing the sign of each $\hat{\delta}_{jir}$ with $\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^{k*}_i - \lambda^{k*}_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^{k*} - \lambda^{k*}_i)\Big) \cdot \hat{\delta}_{jir} < 0$ to its opposite. Thus, we obtain that $\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^{k*}_i - \lambda^{k*}_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^{k*} - \lambda^{k*}_i)\Big) \cdot \hat{\delta}_{jir} \ge 0$ for all $(j,i) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. Accordingly, model $\mathbf{S}_1$ is equivalent to the following maximization LP, denoted as model $\mathbf{S}_2$:

$$
\begin{aligned}
[\mathbf{S}_2]\ \max \quad & \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big\{ \Big[\Big| \hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^{k*}_i - \lambda^{k*}_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^{k*} - \lambda^{k*}_i)\Big)\Big| - \hat{\rho}\Big] \cdot \delta^+_{jir} \Big\} \\
s.t. \quad & 0 \le \delta^+_{ijr} \le 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\}
\end{aligned}
$$

From any optimal solution $\boldsymbol{\delta}^+$ of model $\mathbf{S}_2$, we can derive an optimal solution of model $\mathbf{S}_1$ by setting $\delta_{jir} = \delta^+_{jir}$ if $\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^{k*}_i - \lambda^{k*}_i + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^{k*} - \lambda^{k*}_i)\Big) \ge 0$, and setting $\delta_{jir} = -\delta^+_{jir}$ if $\hat{\tau}_{jir}\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^{k*}_i - \lambda^{k*}_i + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^{k*} - \lambda^{k*}_i)\Big) < 0$, for each $(j,i) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$, so that their objective function values are the same.

For model $\mathbf{S}_2$, its constraint matrix associated with $\delta^+_{ijr} \le 1$ for all $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$ is totally unimodular, as it contains only one entry of 1 in each column. This implies that the feasible solution region of model $\mathbf{S}_2$ is an integral polytope. Thus, there exists an integral optimal solution to model $\mathbf{S}_2$ with $\delta^+_{ijr} \in \{0,1\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. This implies that there exists an optimal solution $\boldsymbol{\delta}$ to model $\mathbf{S}_1$ with $\delta_{ijr} \in \{-1,0,1\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. Hence, there exists an optimal solution to the optimization model defined in (4.31)–(4.34) that satisfies $\delta_{ijr} \in \{-1,0,1\}$ for all $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. Proposition 4.3 is established. □

B.7. Proof of Proposition 4.4

Proof. We first note that the nonlinear model defined in (4.30) for $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\hat{\rho})$ can be reformulated written as the nonlinear optimization model defined in (4.31)–(4.34). We then prove Proposition 4.4 by showing that the nonlinear optimization model defined in (4.31)–(4.34) can be equivalently written as the MILP model defined by (4.35)–(4.39).

Similar to our MILP reformulation of $F_{RP}(\boldsymbol{x},\boldsymbol{z})$, by Proposition 4.3, constraints (4.34) can be replaced with $\delta_{ijr} \in \{-1,0,1\}$ for all $(i,j) \in \mathcal{A}$ and $r \in \{1,2,...,|\mathcal{K}|\}$. By (4.33) we have that $\tilde{\tau}_{ijr} \in \{\overline{\tau}_{ijr} - \hat{\tau}_{ijr}, \overline{\tau}_{ijr}, \overline{\tau}_{ijr} + \hat{\tau}_{ijr}\}$, which, together with $\overline{\tau}_{ijr} \in \mathbb{N}_{>0}$, $\hat{\tau}_{ijr} \in \mathbb{N}_0$ and $\overline{\tau}_{ijr} > \hat{\tau}_{ijr}$, implies that $\tilde{\tau}_{ijr} \in \mathbb{N}_{>0}$. Moreover, we introduce a new variable $\hat{\varphi}_{jir}$ to represent each nonlinear term $\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big) \cdot \tilde{\tau}_{jir} - \hat{\rho}|\delta_{ijr}|$. We then replace each integer variable δ_{jir} with three new binary variables $\hat{\zeta}_{jir,-1}$, $\hat{\zeta}_{jir,0}$ and $\hat{\zeta}_{jir,1}$, which are used to indicate whether δ_{ijr} equals -1, 0 and 1, respectively. Let $\tilde{\tau}_{ijr,-1} = \overline{\tau}_{ijr} - \hat{\tau}_{ijr}$, $\tilde{\tau}_{ijr,0} = \overline{\tau}_{ijr}$ and $\tilde{\tau}_{ijr,1} = \overline{\tau}_{ijr} + \hat{\tau}_{ijr}$. Accordingly, the following linear constraints can be derived for the newly introduced variables, where M_3 is a sufficiently large constant.

$$
\begin{aligned}
& \hat{\zeta}_{ijr,-1} + \hat{\zeta}_{ijr,1} + \hat{\zeta}_{ijr,0} = 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|\} \\
& \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big)\tilde{\tau}_{jir,\ell} - \hat{\rho}|\ell| - M_3(1 - \hat{\zeta}_{jir,\ell}) \le \hat{\varphi}_{jir}
\end{aligned}
\tag{B.5}
$$

$$\leq \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big)\tilde{\tau}_{jir,\ell} - \hat{\rho}|\ell| + M_3(1-\hat{\zeta}_{jir,\ell}),$$

$$\forall\ (j,i)\in,\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}, \ell\in\{-1,0,1\} \tag{B.6}$$

$$\hat{\zeta}_{ijr,\ell}\in\{0,1\},\quad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}, \ell\in\{-1,0,1\}. \tag{B.7}$$

Thus, the nonlinear optimization model defined in (4.31)–(4.34) for $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\hat{\rho})$ can be reformulated to the following maximization MILP model:

$$\begin{aligned} G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\hat{\rho}) = \max\quad & F_1(\boldsymbol{x},\boldsymbol{y}) - \mathcal{Z} + \sum_{(j,i)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}\hat{\varphi}_{jir} - \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}}(M_1 x^k_{ij})\cdot\eta^k_{ij} \\ & + \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|}[M_1(z^k_{ijr}-1)]\cdot(\theta^k_{ijr}+\xi^k_{ijr}) \\ & + \sum_{k\in\mathcal{K}} e^k\cdot(\gamma^k - \lambda^k_{o^k}) + \sum_{k\in\mathcal{K}} l^k\cdot(\lambda^k_{d^k} - \psi^k) \\ s.t.\quad & (4.6)-(4.13), (B.5)-(B.7) \end{aligned}$$

Hence, Proposition 4.4 is proved.

B.8. Proof of Lemma 4.3

Recall that we slightly abuse the notation to define that $\sigma/\|\mathbf{0}\|_1 = 0$ for $\sigma = 0$, $\sigma/\|\mathbf{0}\|_1 = +\infty$ for $\sigma > 0$, and $\sigma/\|\mathbf{0}\|_1 = -\infty$ for $\sigma < 0$. Consider any $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\in\mathcal{X}$. To prove the first statement of Lemma 4.3, we note that if $F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\mathbf{0})) - \mathcal{Z} > 0$, then

$$\hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \geq \frac{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\mathbf{0})) - \mathcal{Z}}{\|\mathbf{0}\|_1} = +\infty,$$

implying that $\hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = +\infty$.

To prove the second statement, consider the case where $F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\mathbf{0})) - \mathcal{Z} \leq 0$, which implies that $(F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\mathbf{0}) - \mathcal{Z}))/\|\mathbf{0}\|_1 \leq 0$. According to the definition of $\rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, we have that $(F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}_l)) - \mathcal{Z})/\|\boldsymbol{\delta}_l\|_1 \leq \rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ for all $\boldsymbol{\delta}_l\in\mathbb{U}\setminus\{\mathbf{0}\}$. Moreover,

- If $\max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}\} \leq 0$, then
$$\hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \max_{\boldsymbol{\delta}\in\mathbb{U}}\ \frac{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}}{\|\boldsymbol{\delta}\|_1} \leq 0.$$
- Otherwise, $\max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}\} > 0$, implying that there must exist a $\boldsymbol{\delta}^*\in\mathbb{U}\setminus\{\mathbf{0}\}$ with $F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}^*)) - \mathcal{Z} > 0$ and $\hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = (F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}^*)) - \mathcal{Z})/\|\boldsymbol{\delta}^*\|_1 > 0$. By Lemma 4.2 and Proposition 4.3, we can assume without loss of generality that $\delta^*_{ijr}\in\{-1,0,1\}$ for all $(i,j)\in\mathcal{A}$ and $r\in\{1,2,...,|\mathcal{K}|\}$, which, together with $\boldsymbol{\delta}^*\in\mathbb{U}\setminus\{\mathbf{0}\}$, implies that $\|\boldsymbol{\delta}^*\|_1 \geq 1$. Thus, we obtain that
$$0 < \hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \leq \frac{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})^*) - \mathcal{Z}}{1} \leq \max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}\}.$$

Hence, it can be concluded that if $F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\mathbf{0})) - \mathcal{Z} \leq 0$, there must be $\hat{\rho}^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \leq \max\{0, F_1(\boldsymbol{x},\boldsymbol{y}) + \max_{\boldsymbol{\delta}\in\mathbb{U}} F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z}\}$.

Lemma 4.3 is proved. □

B.9. Proof of Lemma 4.4

Consider any $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \in \mathcal{X}$. By Lemma 4.2, if $\rho_l \le \rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, we have that $\max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \rho_l\|\boldsymbol{\delta}\|_1\} = G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l) \ge 0$. Note that $\boldsymbol{\delta}(\rho_l)$ indicates a realization of $\boldsymbol{\delta}$ such that

$$F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z} - \rho_l\|\boldsymbol{\delta}(\rho_l)\|_1 = \max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \rho_l\|\boldsymbol{\delta}\|_1\}.$$

Since $\max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \rho_l\|\boldsymbol{\delta}\|_1\} \ge 0$, we obtain that

$$F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z} - \rho_l\|\boldsymbol{\delta}(\rho_l)\|_1 \ge 0,$$

which implies that

$$\rho_l' = \frac{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z}}{\|\boldsymbol{\delta}(\rho_l)\|_1} \ge \rho_l.$$

Therefore, we obtain that

$$\begin{aligned} G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l') &= \max_{\boldsymbol{\delta}\in\mathbb{U}}\{F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta})) - \mathcal{Z} - \rho_l'\|\boldsymbol{\delta}\|_1\} \\ &\ge F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z} - \rho_l'\|\boldsymbol{\delta}(\rho_l)\|_1 = 0. \end{aligned}$$

Thus, by Lemma 4.2, it follows that $\rho_l' \le \rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. Lemma 4.4 is proved. □

B.10. Proof of Theorem 4.2

To prove the first statement of Theorem 4.2, consider each iteration n of Algorithm 2. Let $\rho_l^{(n)}$ denote the value of ρ_l updated in Step 4. Algorithm 2 solves $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l^{(n)})$ in Step 5, derives its optimal solution $\boldsymbol{\delta}(\rho_l^{(n)})$ of $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l^{(n)})$ by (4.40), and computes the value of $\rho_l'^{(n)}$ from $\boldsymbol{\delta}(\rho_l^{(n)})$ by $\rho_l'^{(n)} = (F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l^{(n)}))) - \mathcal{Z})/\|\boldsymbol{\delta}(\rho_l^{(n)})\|_1$. If Algorithm 2 does not terminate at iteration n, we have that $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l^{(n)}) > 0$, which, together with Lemma 4.2, implies that $\rho_l^{(n)} < \rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. Thus, by Lemma 4.4, we know that

$$\rho_l^{(n)} < \rho_l'^{(n)} \le \rho^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}). \tag{B.8}$$

By the definition of $G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l'^{(n)})$ and $\rho_l'^{(n)}$, we have that

$$G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l'^{(n)}) \ge F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l^{(n)}))) - \mathcal{Z} - \rho_l'^{(n)}\|\boldsymbol{\delta}(\rho_l^{(n)})\|_1 = 0. \tag{B.9}$$

Next, consider each iteration m with $m \ge n+1$. If Algorithm 2 does not terminate at iteration m, we can obtain that

$$\rho_l^{(m)} \ge \rho_l'^{(n)}, \tag{B.10}$$

$$G(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\rho_l^{(m)}) = F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l^{(m)}))) - \mathcal{Z} - \rho_l^{(m)}\|\boldsymbol{\delta}(\rho_l^{(m)})\|_1 > 0, \tag{B.11}$$

$$F_1(\boldsymbol{x},\boldsymbol{y}) + F_{LP}(\boldsymbol{x},\boldsymbol{z},\tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l^{(n)}))) - \mathcal{Z} - \rho_l^{(m)}\|\boldsymbol{\delta}(\rho_l^{(n)})\|_1 \le 0, \tag{B.12}$$

where (B.10) and (B.11) are implied by Step 4 of Algorithm 2 and Lemma 4.2, and (B.12) is implied by (B.9) and (B.10). By (B.8) and (B.10) we obtain that $\rho_l^{(n)} < \rho_l^{(m)}$. This, together with (B.11) and (B.12), implies that $\boldsymbol{\delta}(\rho_l^{(n)})$ and $\boldsymbol{\delta}(\rho_l^{(m)})$ are not equal.

According to Proposition 4.3, each $\boldsymbol{\delta}$ derived by (4.40) satisfies that $\delta_{ijr} \in \{-1, 0, 1\}$ for all $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, \cdots, |\mathcal{K}|\}$. Therefore, as there are a finite number of such $\boldsymbol{\delta}$, Algorithm 2 must terminate in a finite number of iterations. Moreover, when Algorithm 2 terminates, we have that $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho_l) = 0$. By Lemma 4.2, we obtain that Algorithm 2 returns $\rho_l = \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and accordingly, $\boldsymbol{\delta}(\rho_l)$ is the corresponding worst-case scenario for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Hence, the first statement of Theorem 4.2 is proved.

The second statement of Theorem 4.2 follows directly from the property of bisection search. For any $\epsilon > 0$, Algorithm 2 only needs at most $\lceil \log_2((\rho_h^{(0)} - \rho_l^{(0)})/\epsilon) \rceil$ iterations to ensure that $\rho_h - \rho_l \leq \epsilon$. At each iteration of Algorithm 2, by Lemma 4.2 and Lemma 4.4 we note that $\rho_l \leq \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \leq \rho_h$ and $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho_l) \geq 0$. Thus, after $\lceil \log_2((\rho_h^{(0)} - \rho_l^{(0)})/\epsilon) \rceil$ iterations, we have that $\rho_l \leq \rho^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \leq \rho_h \leq \rho_l + \epsilon$, and by $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho_l) \geq 0$ we also have that $(F_1(\boldsymbol{x}, \boldsymbol{y}) + F_{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}}(\boldsymbol{\delta}(\rho_l))) - \mathcal{Z} \geq \rho_l \|\boldsymbol{\delta}(\rho_l)\|_1$. Hence, the second statement of Theorem 4.2 is also proved. □

B.11. Proof of Theorem 4.3

At each iteration of Algorithm 3, UB and LB are updated by solving the corresponding master problem and subproblem, while a new worst-case scenario $\boldsymbol{\delta}$ in $\mathbb{U}$ is obtained and added into the scenario subset Λ. Algorithm 3 stops when $UB = LB$. Accordingly, by following an argument similar to that in the proof of Theorem 4.1, we can show as follows that Algorithm 3 must terminate and return an optimal solution to model RS in a finite number of iterations.

First, we show that Algorithm 1 returns an optimal solution to model RS if it terminates with $UB = LB$. As model RSMILP(Λ) is a relaxation of model RS, the value of LB, which equals the optimal objective value of model RSMILP(Λ), is a valid lower bound on the optimal objective value of model RS. As UB is the worst-case normalized cost deviation of a first-stage solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, it provides a valid upper bound on the optimal objective value of model RS. Thus, when $UB = LB$ is achieved, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ forms an optimal solution to model RS.

Next, we show that Algorithm 1 must terminate with $UB = LB$ in a finite number of iterations. To show this, we note that at each iteration n, if the worst-case scenario $\boldsymbol{\delta}^{(n)}$ identified in Step 3 of Algorithm 1 is not in the current scenario subset Λ, it will be added to Λ. According to Proposition 4.3, $\boldsymbol{\delta}^{(n)}$ satisfies that $\delta_{ijr}^{(n)} \in \{-1, 0, 1\}$ for all $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, \cdot, |\mathcal{K}|\}$, and has a finite number of possible values. Therefore, in a finite number of iterations, $\boldsymbol{\delta}^{(n)}$ identified in Step 3 of Algorithm 1 must be included in the current scenario set Λ. In such a situation, both LB and UB must be equal to the optimal objective value of the current master problem, implying that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ forms an optimal solution to model RS. This completes the proof of Theorem 4.3. □

Appendix C: Acceleration Strategies for C&CG Algorithms

In this section, we illustrate several acceleration strategies employed in our implementation of the newly proposed C&CG algorithms.

C.1. Master Problems: Removing Redundant Variables and Constraints, Imposing Valid Inequality, and Breaking Symmetry

For the master problems solved in our C&CG Algorithms, we can strengthen their formulations by removing redundant variables and constraints, imposing a valid inequality, and breaking their symmetric structure.

First, for each commodity $k \in K$, and for each pair of nodes i' and j' of the network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, let $\underline{\tau}^k(i', j')$ denote the length of the shortest-time path from node i' to node j' under the nominal travel times in the flat network, such that the origin o^k and destination d^k of commodity k are not included in between the start and end nodes of the path. It can be seen that for each arc $(i, j) \in \mathcal{A}$, if $\underline{\tau}^k(o^k, i) + \overline{\tau}_{ij} + \underline{\tau}^k(j, d^k) > l^k - e^k$, then under the nominal travel times, commodity k cannot pass arc (i, j) without violating its earliest time for departure from origin o^k or its due time for arrival at destination d^k. Therefore, in every nominal timely-implementable first-stage solution of the robust CTSNDP, commodity k can pass arc $(i, j) \in \mathcal{A}$ only if the following condition is satisfied:

$$\underline{\tau}^k(o^k, i) + \overline{\tau}_{ij} + \underline{\tau}^k(j, d^k) \leq l^k - e^k. \tag{C.1}$$

Define $\mathcal{K}_{ij}$ as the set of such commodities k that satisfy (C.1). Accordingly, variables and constraints associated with commodity $k \in \mathcal{K} \setminus \mathcal{K}_{ij}$ can be safely eliminated from the master problems for each $(i, j) \in \mathcal{A}$.

Second, consider each arc $(i, j) \in \mathcal{A}$, and each pair of different commodities $k, k' \in \mathcal{K}_{ij}$. If k and k' are consolidated and shipped together through arc (i, j) in a nominal timely-implementable first-stage solution to the robust CTSNDP, then due to the constraints of commodities k and k' on earliest times for departure from their origins and latest times for arrival at their destinations, both of the following conditions must be satisfied:

$$\underline{\tau}^k(o^k, i) + \overline{\tau}_{ij} + \underline{\tau}^{k'}(j, d^{k'}) \leq l^{k'} - e^{k'}, \tag{C.2}$$

$$\underline{\tau}^k(o^{k'}, i) + \overline{\tau}_{ij} + \underline{\tau}^k(j, d^k) \leq l^k - e^k. \tag{C.3}$$

Define $\mathcal{K}^2_{ij}$ to be the set of such commodity pairs $(k,' k')$ that satisfy (C.2) and (C.3) above. Accordingly, (C.4) below can be introduced to the master problems as a valid inequality, prohibiting k and k' from being consolidated for each k and k' that do not satisfy conditions (C.2) and (C.3):

$$z^k_{ijr} + z^{k'}_{ijr} \leq 1, \quad \forall\ (i, j) \in \mathcal{A},\ (k, k') \in (\mathcal{K}_{ij} \times \mathcal{K}_{ij}) \setminus \mathcal{K}^2_{ij},\ r = \{1, 2, ..., |\mathcal{K}| - 1\}. \tag{C.4}$$

Third, to break the symmetric structure of each master problem solved in our C&CG algorithms, we can restrict that for each arc $(i, j) \in \mathcal{A}$, the square sum of commodities' indices included in the r-th consolidation on (i, j), which equals $\sum_{k \in \mathcal{K}} (k \cdot k) z^k_{ijr}$, must be non-decreasing in r. Accordingly, the following inequality can be introduced to the master problems, without changing their optimal objective values:

$$\sum_{k \in \mathcal{K}} (k \cdot k) z^k_{ijr} \geq \sum_{k \in \mathcal{K}} (k \cdot k) z^k_{ijr+1}, \quad \forall\ (i, j) \in \mathcal{A}, r = \{1, 2, ..., |\mathcal{K}| - 1\}. \tag{C.5}$$

C.2. Subproblems: Removing Redundant Variables and Constraints

For the subproblems solved in our C&CG algorithms, we can strengthen their formulations by removing redundant variables and constraints. In the following, we first present it for the RO-C&CG algorithm, and then for the RS-C&CG algorithm.

First, for any given nominal timely-implementable first-stage solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ of model RO with (C.5) satisfied, we can obtain its corresponding flat solution denoted by $(\mathcal{P}(\boldsymbol{x},\boldsymbol{z}),\mathcal{C}(\boldsymbol{x},\boldsymbol{z}))$. Let $P^k(\boldsymbol{x},\boldsymbol{z})$ indicate the corresponding flat path for commodity $k\in\mathcal{K}$, with $\mathcal{N}^k(\boldsymbol{x},\boldsymbol{z})$ and $\mathcal{A}^k(\boldsymbol{x},\boldsymbol{z})$ representing its node sequence and arc sequence, respectively. For each arc $\alpha\in\mathcal{A}$, let $\mathcal{C}^\alpha(\boldsymbol{x},\boldsymbol{z})$ indicate the corresponding set of all non-empty consolidations on arc $\alpha\in\mathcal{A}$. As a result, $|\mathcal{C}^\alpha(\boldsymbol{x},\boldsymbol{z})|$ represents the total number of consolidations on arc α.

In any optimal solution to model $F_{RP}(\boldsymbol{x},\boldsymbol{z})$ defined in (3.29)–(3.40) of the subproblem, only arcs in $\mathcal{A}^k$ and nodes in $\mathcal{N}^k$ can be visited by commodity k, implying that $v_{ij}^k=0$ for all $(i,j)\in\mathcal{A}\setminus\mathcal{A}^k$ and $k\in\mathcal{K}$, and that $w_i^k=0$ for all $i\in\mathcal{N}\setminus\mathcal{N}^k$, $k\in\mathcal{K}$. Since empty consolidations are redundant, we have that $b_{ijr}=0$ for all $(i,j)\in\mathcal{A}$ with $|C_r^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|=0$. As a result, excluding these redundant decision variables and their related constraints will not change the optimal objective value of model $F_{RP}(\boldsymbol{x},\boldsymbol{z})$. Accordingly, we can replace $\mathcal{N}$, $\mathcal{A}$, and $|\mathcal{K}|$ in model $F_{RP}(\boldsymbol{x},\boldsymbol{z})$ defined in (3.29)–(3.40) of the subproblem with their corresponding $\mathcal{N}^k(\boldsymbol{x},\boldsymbol{z})$, $\mathcal{A}^k(\boldsymbol{x},\boldsymbol{z})$ and $|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|$, respectively.

With the redundant variables and constraints of model $F_{RP}(\boldsymbol{x},\boldsymbol{z})$ defined in (3.29)–(3.40) excluded, some variables and constraints of its reformulation defined in (4.19)-(4.20) can also be excluded. The resulting model, denoted by $\mathrm{SRP}_1(\boldsymbol{x},\boldsymbol{z})$, is shown as follows, where $\overline{\mathcal{T}}(\boldsymbol{x},\boldsymbol{z})$ denotes the domain defined by (4.6) – (4.13), (4.21) – (4.24) and (B.4), with $\mathcal{N}$, $\mathcal{A}$ and $|\mathcal{K}|$ replaced by their corresponding $\mathcal{N}^k(\boldsymbol{x},\boldsymbol{z})$, $\mathcal{A}^k(\boldsymbol{x},\boldsymbol{z})$ and $|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|$, respectively:

$$
\begin{aligned}
[\mathrm{SRP}_1(\boldsymbol{x},\boldsymbol{z})]\ F_{RP}(\boldsymbol{x},\boldsymbol{z}) = \max_{(\boldsymbol{\zeta},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\psi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\varphi})\in\overline{\mathcal{T}}(\boldsymbol{x},\boldsymbol{z})} & \sum_{(j,i)\in\mathcal{A}^k(\boldsymbol{x},\boldsymbol{z})}\ \sum_{r=1}^{|\mathcal{C}^{(j,i)}(\boldsymbol{x},\boldsymbol{z})|}\varphi_{jir} \\
& -\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}^k(\boldsymbol{x},\boldsymbol{z})}(M_1x_{ij}^k)\cdot\eta_{ij}^k \\
& +\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}^k(\boldsymbol{x},\boldsymbol{z})}\ \sum_{r=1}^{|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|}[M_1(z_{ijr}^k-1)]\cdot(\theta_{ijr}^k+\xi_{ijr}^k) \\
& +\sum_{k\in\mathcal{K}}e^k\cdot(\gamma^k-\lambda_{o^k}^k)+\sum_{k\in\mathcal{K}}l^k\cdot(\lambda_{d^k}^k-\psi^k).
\end{aligned}
$$

As a result, the RO-C&CG algorithm can solve the subproblem for any given nominal timely-implementable first-stage solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ by solving the $\mathrm{SRP}_1(\boldsymbol{x},\boldsymbol{z})$ model. In this model, the number of consolidation indices on each arc $(i,j)\in\mathcal{A}$ is equal to $|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|$, which is generally much smaller than $|\mathcal{K}|$. From the optimal solution obtained for $\mathrm{SRP}_1(\boldsymbol{x},\boldsymbol{z})$, we can compute the worst-case scenario $\boldsymbol{\delta}\in\mathbb{U}(\Gamma)$ for $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ according to (C.6) below, thereby still ensuring the convergence of the RO-C&CG method.

$$
\delta_{ijr}=\begin{cases}-\zeta_{ijr,-1}+\zeta_{ijr,1}, & \text{if } r\in\{1,...,|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|\},\\ 0, & \text{otherwise,}\end{cases}\qquad \forall\ (i,j)\in\mathcal{A}, r\in\{1,...,|\mathcal{K}|\}. \tag{C.6}
$$

Next, for any given nominal timely-implementable first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ of model RS with (C.5) satisfied, consider the maximization MILP model defined in (4.35)–(4.39) for the computation of $\mathrm{G}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho)$ for the subproblem of the RS-C&CG algorithm. By following an argument similar to that above, it can be shown that, to compute $\mathrm{G}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho)$, we also only need to solve model $\hat{\mathrm{G}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho)$ shown below, where $\underline{\mathcal{T}}(\boldsymbol{x}, \boldsymbol{z})$ denotes the domain defined by (4.6) – (4.13) and (B.5) – (B.7), with $\mathcal{N}$, $\mathcal{A}$ and $|\mathcal{K}|$ replaced by their corresponding $\mathcal{N}^k(\boldsymbol{x}, \boldsymbol{z})$, $\mathcal{A}^k(\boldsymbol{x}, \boldsymbol{z})$ and $|\mathcal{C}^{(i,j)}(\boldsymbol{x}, \boldsymbol{z})|$, respectively:

$$
\begin{aligned}
[\hat{\mathrm{G}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho)] \quad \max_{(\hat{\boldsymbol{\zeta}}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \hat{\boldsymbol{\varphi}}) \in \underline{\mathcal{T}}(\boldsymbol{x}, \boldsymbol{z})} \quad & F_1(\boldsymbol{x}, \boldsymbol{y}) - \mathcal{Z} + \sum_{(j,i) \in \mathcal{A}^k(\boldsymbol{x}, \boldsymbol{z})} \sum_{r=1}^{|\mathcal{C}^{(j,i)}(\boldsymbol{x}, \boldsymbol{z})|} \hat{\varphi}_{jir} - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}^k(\boldsymbol{x}, \boldsymbol{z})} (M_1 x_{ij}^k) \cdot \eta_{ij}^k \\
& + \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}^k(\boldsymbol{x}, \boldsymbol{z})} \sum_{r=1}^{|\mathcal{C}^{(i,j)(\boldsymbol{x}, \boldsymbol{z})}|} [M_1(z_{ijr}^k - 1)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k) \\
& + \sum_{k \in \mathcal{K}} e^k \cdot (\gamma^k - \lambda_{o^k}^k) + \sum_{k \in \mathcal{K}} l^k \cdot (\lambda_{d^k}^k - \psi^k).
\end{aligned}
$$

From the optimal solution obtained for $\hat{\mathrm{G}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \rho)$ above, we can compute the worst-case scenario $\boldsymbol{\delta} \in \mathbb{U}$ for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ according to (C.7) below, thereby also ensuring the convergence of the RS-C&CG method.

$$
\delta_{ijr} = \begin{cases} -\hat{\zeta}_{ijr,-1} + \hat{\zeta}_{ijr,1}, & \text{if } r \in \{1, ..., |\mathcal{C}^{(i,j)}(\boldsymbol{x}, \boldsymbol{z})|\}, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}. \tag{C.7}
$$

C.3. Iterations: Bundling New Scenarios to Add

To further enhance the efficiency of both the RO-C&CG and RS-C&CG algorithms, we also implement a bundle strategy to update the scenario set in each iteration. This approach is similar to the one used by Remli et al. (2019) in their Benders decomposition-based algorithm for a transpiration service procurement problem.

In each iteration of our C&CG algorithms, we solve the master problem to obtain an optimal first-stage solution, as well as a pool of feasible first-stage solutions. We can accomplish this using general optimization solvers such as Gurobi and CPLEX. These first-stage feasible solutions, including the optimal solution, are sorted by their objective values in non-decreasing order. For each of these solutions, we then solve the corresponding subproblem to identify its worst-case scenario, resulting in multiple new scenarios that can be added to the master problem for future iterations. As we add more new scenarios in each iteration, the C&CG algorithm may require fewer iterations to reach the optimum solution. However, as we add new scenarios along with their decision variables and constraints, the size of the master problem increases, which may lead to longer computation times for each iteration of the algorithm.

Accordingly, to strike a better balance between efficiency and accuracy, we add a bundle of at most two new scenarios to the master problem in each iteration. One of these new scenarios is the worst-case scenario of the optimal first-stage solution. To identify the second scenario to add, we evaluate the first-stage solutions in the pool and choose the solution with the least objective value. If this solution's objective value is better than the current best upper bound on the optimal objective value, we update the upper bound accordingly, and we then add the worst-case scenario of this best solution, as the second scenario in the bundle, to the master problem.

Appendix D: Details on Instance Generation for Computational Experiments

For our computational experiments, we generated instances of the robust CTSNDP based on the seven instance classes (named R4-R10) of the fixed-charge capacitated multi-commodity network design (CMND) problem available in the literature Ghamlouche et al. (2003). The attributes of each instance class, including the size of the node set $|\mathcal{N}|$, the size of the arc set $|\mathcal{A}|$, and the size of the commodity set $|\mathcal{K}|$, are given in columns 2-4 of Table 5.1. Each class consists of five networks with varying values for the ratio of fixed cost to variable cost, and for the ratio of the total demand of commodities to the total capacity of the network. These instances are referred to as "untimed" instances, as they do not have any temporal attributes such as travel times of arcs, or earliest available times and due times of commodities.

To obtain "timed" instances for each of the 7 classes of the CMND problem, we followed an approach similar to that presented in Boland et al. (2017) to generate fixed costs and time attributes of the CTSNDP. First, for each arc $(i,j) \in \mathcal{A}$, we set the nominal value of travel time (in minutes) $\overline{\tau}_{ij}$ to be proportional to its fixed cost f_{ij} by setting $\overline{\tau}_{ij} = f_{ij}/0.55$, as in Boland et al. (2017). This is based on the same premise that f_{ij} represents the transportation cost for carriers that spend 0.55 cents per mile, and that their trucks travel at 60 miles per hour.

Next, for each commodity $k \in \mathcal{K}$, we followed a normal distribution to randomly generate the available time e^k. Let $\mathscr{L}_k$ denote the length of the shortest-time path from origin o^k to destination d^k for commodity k in the flat network under the nominal travel times $\overline{\boldsymbol{\tau}}$. We then set the due time l^k of each commodity $k \in \mathcal{K}$ by $l^k = e^k + \mathscr{L}_k + \mathscr{F}_k$. Here, the parameter $\mathscr{F}_k \geq 0$ represents the time flexibility for the delivery of commodity k, which we also set randomly using a normal distribution. We used the same normal distribution to generate the available times e^k for all instances, but used two different normal distributions to generate $\mathscr{F}_k$ for instances of high and low time flexibility, respectively. Consequently, we had two combinations of normal distributions to generate commodities' available times and time flexibility. The detailed settings of these normal distributions are described in Table D.1, where $\mathscr{L}$ denotes the average of $\mathscr{L}_k$ over all $k \in \mathcal{K}$.

Table D.1 Detail setting of the normal distributions used for generating "timed" instances.

Normal Distribution	Mean(μ)	Standard Deviation(σ)
For generating e_k	$\mathscr{L}$	$\frac{1}{6}\mathscr{L}$
For generating $\mathscr{F}_k$	$\frac{1}{2}\mathscr{L}$	$\frac{1}{6} \cdot \frac{1}{2}\mathscr{L}$
	$\frac{1}{4}\mathscr{L}$	$\frac{1}{6} \cdot \frac{1}{4}\mathscr{L}$

For each "timed" instance obtained, we then generated unit in-storage holding costs and unit delay penalties for the commodities. We set the per-unit-of-demand-and-time in-storage holding cost h^k for each commodity $k \in \mathcal{K}$ to be proportional to its cheapest per-unit-of-time per-unit-of-flow cost, i.e., $h^k = 0.5 \min_{a \in \mathcal{A}}\{(c_a^k + f_a/u_a)/\overline{\tau}_a\}$ where $\overline{\tau}_a$ is the nominal value of the travel time generated. Inspired by Lanza et al. (2021), for each commodity $k \in \mathcal{K}$, we set its penalty g^k per unit of time for the delay to be twice the most

expensive per-unit-of-time transportation cost for it to pass through an arc, i.e., $g^k = 2 \cdot \max_{a \in \mathcal{A}}\{(c_a^k \cdot q^k + f_a\lceil q^k/u_a \rceil)/\overline{\tau}_a\}$.

Moreover, to characterize travel time uncertainty, we generated the maximum deviation $\hat{\tau}_{ij}$ of the travel time for each arc $(i,j) \in \mathcal{A}$ by setting $\hat{\tau}_{ij} = \hat{\mu}_{ij}\overline{\tau}_{ij}$. Here, $\overline{\tau}_{ij}$ is the nominal value of the travel time generated, and $\hat{\mu}_{ij}$ is a coefficient randomly selected from 0.1 to 0.5.

For each network in each problem class, we randomly generated 3 instances for each combination of the distributions for commodities' available time and time flexibility. As a result, we obtained $5 \times 2 \times 3 = 30$ test instances for each of the 7 instance classes, and thus obtained $7 \times 30 = 210$ test instances in total.