CONTINUOUS-TIME SERVICE NETWORK DESIGN: NEW MODELS, RELAXATIONS, AND SOLUTION METHODS

SHU SHENGNAN

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University
Department of Logistics and Maritime Studies

Continuous-Time Service Network Design: New Models, Relaxations, and Solution Methods

Shu Shengnan

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

April 2022

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: ____________________

Name of Student: Shu Shengnan

Acknowledgements

My PhD studies have been an ongoing life-changing experience, one full of challenges, both academically and personally. I would like to express my gratitude to all those who have helped me during this long and arduous journey. Without their help and encouragement, I would not have been able to pluck up my courage to complete this journey.

First and foremost, I would like to express my sincere appreciation to my PhD supervisor, Professor Zhou Xu, for the opportunity to learn from him, as well as for his valuable suggestions, patient guidance and constant encouragement during my PhD studies. I greatly admire his expertise, rigor, and enthusiasm as an academic, and also respect his personality and diligence, as well as his timely availability to provide valuable feedback for my studies and firm support when I get stuck. It has been my honor to benefit from his guidance and supervision both academically and personality-wise, and is something that I will treasure for the rest of my life.

I would also like to thank Professor Roberto Baldacci for his instruction and help throughout my studies and work. It was truly an honor for me to take his graduate courses and then to collaborate with him in my research work. His professional and academic enthusiasm have greatly impressed me. I have gained invaluable knowledge, techniques, and experience from him, and his support has been vital for the completion of my thesis.

During my PhD studies, I have also had the great honor of research collaboration with Professor Chung-Yee Lee. It has been such a rewarding and joyful experience to work with Professor Lee, and his advice has been invaluable to me.

I wish to thank the wonderful faculty members for their instruction and assistance during my studies at the Hong Kong Polytechnic University. In particular, I greatly appreciate Professor Pengfei Guo, Professor Li Jiang and Professor Miao Song for their excellent PhD courses. I would also like to acknowledge the assistance of the kind staff in the Department of Logistics and Maritime Studies, especially Ms Irene Lam, Ms Lorraine Leung and Ms Anne Wong.

My special thanks are also extended to Professor Zhixing Luo, Professor Hu Qin and Professor Qinghua Wu, who first introduced me to the field of operation research during my master's studies, and then continued to support me during my PhD studies. Their suggestions and guidance strongly encouraged me to pursue a PhD in this field. I am very grateful for the significant influence that they have had on my academic career.

Moreover, I am grateful to Professor Qiang Meng, Professor Jin Qi, and Professor Shuaian Wang for serving on my thesis committee, and for their important and insightful suggestions on my thesis.

Finally, I would like to express my deepest appreciation to my family and friends for their constant support and encouragement.

Abstract

This thesis aims to enrich both deterministic and robust optimization techniques for the continuous-time service network design problem (CTSNDP), which occurs widely in practice. It consists of two studies, with the first one incorporating holding costs and the second one further considering uncertain travel times.

The CTSNDP is to minimize the total operational cost for consolidation carriers by optimizing the schedules of transportation services and the routes of shipments for dispatch, which can occur at any time point along a continuous-time planning horizon. In order to be cost effective, shipments often wait to be consolidated, which incurs holding costs. Holding costs not only contribute to the overall total cost, but also affect the decisions on routing and consolidation plans in the CTSNDP. Despite their importance, holding costs have not been considered in existing exact solution methods for the CTSNDP, since incorporating them significantly complicates the problem. The correctness of all these methods relies on the assumption of zero holding costs. To tackle this challenge, the first study of this thesis develops a new dynamic discretization discovery (DDD) algorithm, based on the typical time-expanded network, that can solve the CTSNDP with holding costs (CTSNDP-HC) to exactly optimum. The algorithm is based on a novel relaxation model, a new upper bound heuristic, and a new discretization refinement procedure. Results of computational experiments demonstrate the effectiveness and efficiency of our proposed algorithm, both in finding optimal solutions to the CTSNDP-HC and in producing tight lower and upper bounds. The experimental results also show the benefits of considering holding costs in the CTSNDP-HC.

Due to various uncertainty factors, such as weather and traffic conditions, actual travel times often fluctuate. Travel time uncertainty is thus a vital source of variability in the CTSNDP-HC. This motivates the second study of this thesis on the robust CTSNDP-HC. With uncertain travel times incorporated, solutions to the robust CTSNDP-HC can lead to service network designs that not only provide reliable services to transit shipments, but also minimize their operational costs. However, the time-expanded network used in modeling and solving the deterministic CTSNDP-HC turns out to be inappropriate for incorporating uncertain travel times in the robust CTSNDP-HC. To address this challenge, a new deterministic optimization model for the CTSNDP-HC based on the physical network is newly proposed. This new model formulates the time component of the CTSNDP-HC by a set of variables and constraints, with their indices indicating shipment consolidations. Based on this, we derive a two-stage robust optimization model for the robust CTSNDP-HC, using a probability-free budgeted uncertainty set to incorporate uncertain travel times. To solve the robust CTSNDP-HC, we apply a classical column-and-constraint generation (C&CG) method, and then enhance the method via some novel optimization techniques of dynamic parameter adjustment. Results of computational experiments demonstrate the effectiveness and efficiency of the two-stage robust optimization model and the enhanced C&CG solution method, as well as the benefits of incorporating uncertain travel times in the robust CTSNDP-HC

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

With the rapid development of cross-region trading, especially for e-commerce, shipment consolidation has become vital in the freight transportation industry. Particularly in less-than-truckload (LTL) transportation, where each shipment is relatively small compared to the capacity of trailers, carriers often need to consolidate the shipments for cost effectiveness when they are deciding how to route the shipments through a network of terminals. At each terminal visited, a shipment may be unloaded from an inbound trailer and then loaded onto an outbound trailer, whereby trailers can be shared with other shipments. Shipment consolidation is challenging, as it requires coordinating the transportation of various shipments in both space and time. Accordingly, a freight carrier has to decide on the routing and consolidation plans for the shipments, and also has to design the shipping services for the execution of the plans, so that all service standards are met for the routing of the shipments, with the total cost also being minimized. This results in the well-known *Service Network Design Problem* (SNDP) [33, 109].

Continuous Time and Holding Costs

The Service Network Design Problem (SNDP) has for decades since the 1990s been one of the most widely studied problems in the operations research community [38, 47], due to its rich practical applications as well as theoretical significance.

Decisions in the SNDP have both temporal and spatial components, as when and at which terminals to pick up a certain commodity for a certain service both need to be determined. It is known that the SNDP is strongly NP-Hard [56], and thus it is unlikely that there exists a polynomial time algorithm that can always produce an exact optimal solution to the SNDP in polynomial time. To explore heuristic solutions to the SNDP, the temporal component of the problem is often modeled approximately by a discretization method [72, 2, 44, 37]. Through discretization, the time horizon is split into several time intervals, so that instead of stating the exact departure time (e.g. 7:10 pm), the time interval (e.g., between 7 pm and 8 pm) can be determined, during which a commodity may be dispatched from a terminal. As a result, the SNDP can be modeled and solved approximately based on a discretized *time-expanded network*, in which every node represents a pair of a geographic terminal and a discrete time interval, and every arc that links two nodes represents a shipping service with departure time and transit duration defined by the time interval of the two nodes [2, 50, 51, 86].

The discretized time-expanded network provides a flexible modeling paradigm for the SNDP, that is, a time-index (TI) formulation, and this makes it much easier to develop solution methods. However, in such solution methods that are based on a discretized time-expanded network, it is challenging to choose an appropriate time discretization level that directly impacts the solution difficulty and quality. Such solution methods also fail to guarantee to produce a continuous-time optimal solution to the SNDP, leaving two research questions to answer: (1) Does there always exist a finite fully discretized time-expanded network that is sufficient for the solution methods to generate a continuous-time optimal solution to the SNDP? Boland and Savelsbergh [24] noted that for some problems, the existence of such a fully discretized time-expanded network is not straightforward. For these kinds of problems, such as the continuous-time inventory routing problem, the optimal discretization level of the network may be smaller than 1 and is difficult to identify [75]. (2) If a finite fully discretized time-expanded network exists, how can the continuous-time optimal solution to the SNDP be produced efficiently? Boland et al. [20] noted in their paper that for a week-long planning horizon, there will be 2,016 time copies for each physical terminal under a 5-minute discretization level, for which the resulting discretized time-expanded network is too large to be applied in the development of efficient solution methods. Ford Jr and Fulkerson [50, 51] and Skutella [94] also mentioned that the size of the time-expanded network depends on the granularity of the time discretization and the number of discretized time points, and that it can grow exponentially with the problem size, significantly impacting the computational tractability. Hence, solving continuous-time optimal solutions for the SNDP based on a fully discretized time-expanded network is well known to be computationally challenging. This motivates studies on the development of efficient solution methods for the *Continuous-Time Service Network Design Problem* (CTSNDP).

Solutions to the CTSNDP indicate accurate departure times and consolidation plan for shipments. Boland et al. [20] proved the existence of a finite fully discretized time-expanded network for the CTSNDP, and they proposed a *Dynamic Discretization Discovery* (DDD) algorithm to solve the optimal solution for the CTSNDP. This DDD algorithm is based on a *partially time-expanded network* that contains only a subset of the time points of the fully time-expanded network, and where arcs are constructed to hold some expected properties. The DDD algorithm repeatedly solves an SNDP on a partially time-expanded network, and refines the partially time-expanded network by following certain refinement strategies. The algorithm stops when the optimal solution obtained from the SNDP can be converted to an optimal solution of the CTSNDP. Marshall et al. [81] further extended this DDD scheme for the CTSNDP based on an interval-based network. However, all these studies on the DDD algorithm and its extension [20, 81] assume that holding freight at a terminal incurs no additional cost. They claim that many carriers operate their own holding terminals where holding commodities will not increase the total cost, and that if the carriers use third-party-owned holding terminals, the transportation savings achieved by the consolidations can fully cover the holding costs. The correctness of their DDD algorithms for the CTSNDP rely on this zero holding costs assumption, as these methods depend on a special solution structure that is only valid when the holding costs are zero [20, 62].

However, the holding costs do indeed play an important role in the CTSNDP, and thus need to be incorporated. First, in practice, holding commodities in terminals always incurs additional costs, adding to the total cost. Even in carriers' self-owned holding facilities at terminals, there are some operating costs, such as rental and labor costs, which need to be shared among the commodities. The holding costs are often different in different terminals and for different commodities [70, 99, 88]. It has been widely noted that consolidations cause longer delivery times, and hence induce holding costs [61]. Many existing studies on shipment consolidation take into account holding costs [98, 100, 25, 88, 70, 99], which are also referred to as a *consolidation penalty cost* [100]. Second, although some studies claim that the freight holding costs often make up less than 5% of the total operating cost, for LTL carriers who often spend millions of dollars on transportation in a week, a small percentage improvement in operating costs means a significant amount of monetary saving, perhaps meaning the difference between profitability and losing money. Furthermore, as observed by Rudi et al. [88] in their study on the capacitated multi-commodity network flow problem, the choices of transport modes with holding costs taken into account are significantly different from the choices that are made without considering the holding costs, resulting in different degrees of increase in the total cost of the latter's choices. Thus, it is of great interest to investigate, as is done in this thesis, how the holding costs affect the optimal total operating cost as well as the optimal decisions on shipping service design, shipment transportation, and shipment consolidations, in the CTSNDP.

Due to their importance, holding costs have been taken into account in some existing studies on the SNDP and its variants [4, 3, 86, 63]. However, all solution methods proposed in these studies are based on the discretization scheme. Similar to the SNDP, the solution methods based on the discretization scheme for the SNDP with holding costs cannot guarantee to produce continuous-time optimal solutions. Despite the importance of the holding costs in the SNDP, neither the existence of the finite fully discretized time-expanded network nor the efficient exact solution method for the continuous-time SNDP with holding costs are known in the literature. As we mentioned earlier, the up-to-date existing methods proposed for finding the optimal solutions to the CTSNDP [20, 81] are not valid when the holding costs are positive. Marshall et al. [81] noted in their paper that it is a challenging extension for the CTSNDP when it is no longer free to hold freights for consolidation. To incorporate holding costs, one needs to optimize the decisions on locations and times for holding the commodities, which significantly complicates the problem. These motivate the first study of this thesis to develop the first exact solution method for the *Continuous-Time Service Network Design Problem with Holding Costs* (CTSNDP-HC).

Uncertainty of Travel Times

Classical deterministic optimization models assume that all problem parameters are known perfectly in advance. However, in many real-life problems, such as the SNDP, several critical parameters, such as travel times and demands, are uncertain. Hence, without incorporating such uncertainties, deterministic optimization models cannot reflect the actual dynamic behavior of the uncertain parameters of these complex real-life problems. Moreover, decision-makers often prefer *robust* solutions that exhibit good performance over all the possible realizations of the uncertain parameters [74].

One of the common approaches to model uncertainty is to apply stochastic programming, which was introduced by Dantzig [39] in 1955. This approach assumes that the probability distribution of the uncertain parameters is known in advance, and the decision-maker minimizes/maximizes the expected objective value over a set of possible realizations of the uncertain parameters. Despite the great influence and theoretical impact of stochastic programming [89, 19, 43], the limitations of stochastic programming are obvious [83, 90, 30, 8]. First, the precise distribution of the uncertain parameters is hard to be identified, whereas the result of the stochastic programming is sensitive to the considered distribution. Second, stochastic programming models are often based on a given set of possible realizations of uncertain parameters, referred to as scenarios. The solution quality of the stochastic programming relies on the completeness of the considered scenario set, whereas the size of the stochastic programming model usually increases with respect to the number of scenarios considered. To find the best solution, especially for large-sized instances, it is necessary to incorporate a large number of scenarios, which often makes the stochastic programming models hard to solve. Moreover, stochastic programming models are only concerned with the average performance of the system by minimizing the expected total cost, and hence they are powerless to handle risk aversion.

In view of such limitations of stochastic programming, another common approach, named robust optimization [96, 10, 11, 8], can be applied to model uncertainty. Robust optimization assumes a probability-free uncertainty set and optimizes for the worse-case realization over all possible realizations within the uncertainty set. It only requires certain modest assumptions about the probability distributions of the uncertain parameters, and can preserve computational tractability of the deterministic problem. For example, compared to precise probability distribution, it is often much easier to identify lower/upper bounds or discrete collections of possible values for the uncertain parameters. Utilizing bounds on the uncertain parameters to construct the uncertainty set, various robust optimization approaches have been developed, such as the robust optimization approaches based on the box uncertainty set [96, 11, 71], ellipsoidal uncertainty set [10, 17, 71] and polyhedral uncertainty set [16, 12, 71]. To reduce the conservatism of robust optimization solutions due to optimizing for the worst-case scenario, the adjustable/adaptive robust optimization methods [9, 97, 14] with multiple decision stages, and the budget-of-uncertainty approaches [16, 15, 29], which control how much the uncertain parameters can deviate from their nominal values, are widely utilized. More recently, distributionally robust optimization has been used to tackle optimization problems under distribution uncertainty [26, 58, 41]. In contrast with uncertainty-set based robust optimization, which plans against the worst-case realization in the uncertainty set, distributionally robust optimization plans against the worst-case distribution in the ambiguity set of the feasible candidate distributions.

Most of the existing studies that incorporated uncertainties in service network design focus only on demand uncertainty. These studies mainly applied either a scenario-based stochastic programming approach to optimize for the average performance over a set of possible demand realizations [79, 66, 6], or they applied a robust optimization approach to optimize for the worst-case demand realization in a probability-free uncertainty set [108, 5, 73]. For all the existing methods that incorporated the travel time for the SNDP with demand uncertainty, the classic TI formulation of the deterministic SNDP based on a discretized time-expanded network was extended [79, 66, 6, 108].

In addition to the demand, travel time is also an important source of uncertainties in the design of service networks. Due to various uncertain factors, such as weather and traffic conditions, actual travel times fluctuate and thus are unknown to carriers when they design service networks. Uncertain travel times often cause delays to transportation services in actual operation, so that commodities may not be delivered on time. Such disruptive impacts will result in a penalty cost for late delivery or may require some additional express services, which are very costly. With uncertain travel times incorporated, the robust solutions can not only provide reliable services for transit shipments, but also minimize their operational costs. As the uncertainty of travel time is so critical to the SDNP, its disruptive impacts on actual operation need to be taken into account, which further complicates the optimization problem.

Despite its great importance, travel time uncertainty has seldom been taken into account in studies of the SDNP due to its complexity. In addition, the few existing studies that have considered travel time uncertainty in the SNDP have the following limitations, which have motivated the second study of this dissertation.

First, all existing studies that have incorporated travel time uncertainty in the SNDP had only limited consideration of delay propagation caused by consolidations. In the SNDP, any late delivery of a shipment is caused not only by its own transportation delays, but also by the transportation delays of other shipments that need to be consolidated with it. However, most the existing studies failed to take into account the disruptive impact of delays caused by consolidations [78, 114, 76]. Those few studies that considered the consolidation delay propagation restricted each service to only being used at most once within the whole time horizon, which makes the problem simpler than the classical SNDP defined over the time-expanded network [42, 68].

Second, for those existing studies of the SNDP that incorporated travel time uncertainty but had restricted considerations of delay propagation, only a scenario-based stochastic programming model and a simulation model have been adopted [42, 68]. These studies required complete probability information about travel time realizations, which is difficult to know in advance. As a result, only approximation or heuristic methods are proposed for solving the models, thus limiting their usefulness in practice. As far as we know, no existing studies of the SNDP under travel time uncertainty have adopted a robust optimization solution framework. The main challenge is that the common TI formulation based on the time-expanded network turns out to be ineffective for developing robust optimization models and their solution methods for the SNDP under travel time uncertainty. In order to model uncertain travel times, it is necessary to take into account every possible realization of travel times. More precisely, to deal with travel time uncertainty, the time-expanded network based on a discrete-time planning horizon needs either to contain service arcs with all possible travel times, or to vary with different realizations of travel times. In either case, the resulting models are difficult to solve.

Third, no existing robust optimization methods tackle travel time uncertainty for the SNDP under a continuous-time planning horizon and with non-zero holding costs. Even for demand uncertainty, those solution methods known in the existing literature cannot deal with the continuous-time planning horizon or the non-zero holding costs. For modeling and solving the SNDP under travel time uncertainty, the case under a continuous-time planning horizon is much more complicated than the case under a discrete-time planning horizon, with the latter case already being very challenging.

Despite all the difficulties, though, finding an optimal continuous-time robust solution for the problem with non-zero holding costs is of great practical value.

Motivated by the importance of travel time uncertainty in service network design and the limitations inherent in existing studies, the second study of this thesis aims to develop an effective robust optimization formulation and efficient solution methods for the CTSNDP-HC under travel time uncertainty.

1.2 Literature Review

Service network design problems have been widely studied in the literature since the 1990s [38, 47] due to the wide range of applications they cover [33, 109]. An early classification distinguishes between *deterministic* and *robust* service network design problems, where in the robust variant the uncertainties of parameters are highlighted. In this section, we review the existing literature on these two classes of service network design problems, respectively.

1.2.1 Deterministic Service Network Design Problems

For a review of the deterministic variants and associated applications, the reader is referred to Crainic [33] and Wieberneit [109]. Below, we first focus on works that are closely related to the deterministic SNDP and to the solution approaches based on discretization methods, and we then refer to the literature on service network design problems with the objective of capturing consolidation costs such as in-storage holding costs.

Discretization methods

The literature shows different discretization methods aimed at deriving relaxations of TI models for routing and scheduling problems with time constraints. Wang and Regan [106] and Wang and Regan [107] proposed relaxations of TI formulations for a vehicle routing problem and for the Traveling Salesman Problem with Time Windows

(TSPTW), respectively. The relaxations are obtained by partitioning the time windows into a collection of nonoverlapping time intervals, and by defining variables associated with these time intervals. The resulting relaxations are exploited to derive some strong cutting planes that are embedded in a branch-and-cut solution framework. A similar relaxation, called *time bucket relaxation*, was investigated by Dash et al. [40] to solve the TSPTW by a branch-and-cut algorithm that also makes use of valid inequalities derived from the bucket formulation.

Boland et al. [20] introduced a DDD algorithm to solve the optimal solution for the CTSNDP. The DDD algorithm solves a sequence of MIPs defined on a subset of times (i.e., a partial discretization), with variables indexed by times in the subset, that provides lower bounds on the optimal continuous-time value. At each iteration of the algorithm, new times are discovered and used to refine the partial discretization. Once the right subset of times is discovered, the resulting MIP yields the continuous-time optimal value. As highlighted by Boland et al. [20], the refinement strategies of Wang and Regan [107] and Dash et al. [40] employed a DDD algorithm as a preprocessing scheme, rather than a dynamic nonuniform scheme. Further, Boland et al. [20] also focus on the size of the partially time-expanded network by keeping the number of time points in the network to a minimum. The recent work of Marshall et al. [81] further extends that of Boland et al. [20] by modeling the discretization in terms of time intervals instead of time points. This new discretization leads to more effective and efficient DDD algorithms. The algorithm of Marshall et al. [81] can handle larger instances involving up to 30 nodes, 685 arcs, and 400 commodities, and can generate high-quality solutions more quickly than that of Boland et al. [20].

Solution methods based on the DDD solution framework for service network design problems were also investigated by Hewitt [62] and Medina et al. [82]. Hewitt [62] considered variants of the service network design problem encountered in the LTL freight transportation industry. They both proposed multiple enhancements to the DDD algorithmic framework based on inequalities and symmetry-breaking branching rules. Medina et al. [82] introduced an optimization problem that integrates long-haul and local transportation planning decisions. The authors proposed a route-based and an arc-based formulation for the problem that are both solved by means of a DDD algorithm. Other applications of the DDD solution framework can be found in Vu et al. [103], whereas for further perspectives on various aspects of time-dependent models and the DDD, the reader is referred to Boland and Savelsbergh [24].

All the aforementioned works disregard holding costs. In particular, as we will show later, the correctness of the DDD approach proposed by Boland et al. [20] strongly relies on the assumption that the holding costs are equal to zero. This motivates our study on the solution method for the CTSNDP with positive holding costs.

Handling consolidation costs

Consolidation is the process of grouping different items that originate at different locations and different times into single vehicle loads at intermediate terminals or facilities, and transportation costs must be weighed against the penalties (such as handling and in-storage holding costs) that come with consolidation [61].

Several works have highlighted the importance of considering consolidation costs in service network design. Ülkü [100] addressed holding costs as consolidation penalty costs, and presented three shipment consolidation policies, namely, time, quantity and hybrid policies. Pedersen et al. [86] focused on a generic model for transportation service network design with asset management considerations. The authors modeled asset positioning and utilization through constraints on asset availability at terminals, with the consideration of in-storage holding costs. The problem was formulated by means of an arc-based model, and a tabu search metaheuristic was used for its solution. Rudi et al. [88] investigated a capacitated multi-commodity network flow model for the planning of intermodal transportation services, with carbon emissions and in-transit holding costs taken into account. They applied the model on a set of industry data and investigated the interrelations between the decision criteria for greenhouse gas emissions, cost, and time, as well as the impact of inventory holding costs. Jarrah et al. [72] and Erera et al. [44] investigated real-world service network design problems faced by LTL freight transportation carriers. Both of the works considered handling costs at intermediate terminals. Jarrah et al. [72] described an IP formulation capturing the different LTL requirements that is solved using a slope scaling and load-planning tree generation method. Erera et al. [44] presented integer linear programming (IP) models and a matheuristic solution approach for large-scale instances that result in practical applications. Additional works analyzing the trade-off between transportation and holding costs can be found in Bookbinder and Higginson [25], Tyan et al. [98], Ulku [99] and Hu, Toriello and Dessouky [70].

Holding costs for the SNDP and its variants formulated using TI models have been considered by several works, such as Andersen et al. [4, 3], Pedersen et al. [86], and Hewitt et al. [63]. However, due to the approximation introduced by the discretization, the solution methods proposed in these works cannot guarantee the optimality of the solutions obtained. To the best of our knowledge, no exact algorithm has been proposed for the CTSNDP-HC, and the related literature is quite scarce. A continuous SNDP with vehicle asset management was investigated by Hosseininasab [67], where vehicle waiting and holding costs were also considered. Belieres [7] considered tactical transportation planning in a multi-product supply chain inspired by the collaboration between a third-party logistics company and a restaurant chain. The problem was formulated using a TI model with holding costs, and it was solved by means of a hybrid matheuristic based on the DDD algorithm, as the author observed that the DDD method proposed by Boland et al. [20] cannot be used directly in the presence of in-storage holding costs. The algorithm was tested on real-world instances, and the results show that refining the granularity of the time discretization generates substantial savings in terms of holding costs. In the first study of this thesis, we develop a novel DDD algorithm that can efficiently solve the CTSNDP-HC to optimality.

1.2.2 Robust Service Network Design Problems

The existing studies mentioned in Section 1.2.1 are all based on the deterministic nature of the SNDP. It is widely known that the SNDP encounters difficulties in handling uncertainties. This motivated another stream of existing studies that took into account uncertainties in the SNDP.

For the SNDP under uncertainties, most of the existing studies have focused on demand uncertainty and have developed either stochastic optimization methods or robust optimization methods. Among the stochastic optimization methods, Lium et al. [79] investigated the significance of incorporating stochastic elements into some periodic SNDP formulations, where demand uncertainty is considered to be one example. Their study revealed that uncertainty plays an important role in the SNDP and that the solutions based on a stochastic optimization approach can be structurally different from their deterministic counterparts. Hoff et al. [66] followed the work of Lium et al. [79] and developed the first metaheuristic method for a real life-sized periodic stochastic SNDP under demand uncertainty. Bai et al. [6] further extended the work of Lium et al. [79] by considering vehicle rerouting in the stochastic SNDP under demand uncertainty. Although it is computationally more expensive to solve, their model has the potential to reduce the costs of network planning and outsourcing. In addition, Hewitt et al. [63] proposed a scenario-based cycle-path stochastic optimization formulation for a scheduled SNDP with resource acquisition and management under demand uncertainty. They presented two solution approaches for this problem, including a column generation-based heuristic and a matheuristic.

To further address the demand uncertainty without full distribution information, Atamtürk and Zhang [5] proposed a two-stage robust formulation for the network flow and network design problem under demand uncertainty, which is a variant of the SNDP without a temporal component. They also compared the computational performances among the two-stage robust optimization method, the single-stage robust optimization method, and the stochastic programming method. Their experimental results indicated that their proposed two-stage robust optimization outperforms both the scenario-based stochastic programming and the more conservative single-stage robust optimization. Koster et al. [73] successfully applied Γ-uncertainty to formulate a robust service network design problem under demand uncertainty, where the service network is static and has no temporal component. They assumed that the worst-case demand realization in the given uncertainty set must satisfy the capacity constraints of all the services, with outsourcing not allowed. Furthermore, Wang and Qi [108] proposed a two-stage robust model for the SNDP under demand uncertainty based on the discretized time-expanded network and with outsourcing allowed. They solved this problem by a column-and-constraint generation method.

In addition to demand uncertainty, travel time uncertainty is also a major source of uncertainties that needs to be taken into account in various transportation problems. Many studies on transportation problems have considered travel time uncertainty, including not only well-known classical transportation problems, such as the vehicle routing problem with time window [69, 1] , shortest path problem [28, 69, 112] and traveling salesman problem [84, 27, 113], but also some specific industry applications, such as airline scheduling [95], pollution routing [46], liner ship schedule design [104, 105] and daily drayage planning [45]. However, these transportation problems are different from the SNDP, as no service designs and no consolidations need to be determined. Thus, no consolidation delays occur in these problems.

For the SNDP and its variants, there is a lack of studies that incorporate travel time uncertainties, for which modeling and solution approaches are limited. Motivated by real-life applications, Yao et al. [110] studied a bus transit route network design problem considering travel time variations. Other than using the average travel time for each service leg, the authors adopted a reliability-based travel time by adding a traveler's risk preference-based buffer time to the average travel time. They proposed a nonlinear optimization model which aims to maximize the efficiency of the passenger trips with the reliability-based travel times taken into account, and they developed a tabu search algorithm to solve the model. Zhao et al. [114] focused on an intermodal service network design problem in a sea-rail transportation system under uncertain travel times, transfer times, and demands, and they introduced chance constraints on both capacity and on-time delivery. However, this study only considered some specific feasible service routes that have several ship services followed by a train service, but it did not take into account consolidation. Liang et al. [78] investigated a bus bridging service design problem for recovery from disruptions in a rail transit system, and formulated it as a path-based multi-commodity flow model without consolidation. Lanza et al. [76] addressed the scheduled SNDP with stochastic travel times, in which each service consists of a sequence of service legs with a scheduled departure time. The authors derived a two-stage mixed-integer linear stochastic programming model defined over a time-expanded network and proposed a progressive hedging-based meta-heuristic to solve it. The first stage of their model determines the selection of services and the routing of the freight flows, while the second stage verifies the delay penalties. However, the authors only considered the delay propagation for service legs within the same service. All these above studies did not consider the delay propagation caused by consolidations.

To the best of our knowledge, there are only two studies that have considered consolidation delay propagation for the continuous-time SNDP under travel time uncertainty. One is Demir et al. [42], which studied an energy-ware intermodal service network design problem with uncertainties in both travel times and demands, where motor-carrier transportation services must catch rail and maritime transportation services according to their fixed schedules. In this study, the authors explored continuous-time robust solutions for the planning of services in the network, and considered the delay propagation caused by transfers and consolidations. The other is Hrušovskỳ et al. [68], which extended the energy-ware intermodal service network design problem by incorporating in-transit holding costs, and formulated the travel time uncertainty by a simulation model.

However, for both of the two studies introduced above [42, 68], services are allowed to be used at most once within the whole planning horizon, which simplifies the model formulation, but limits its applications in practice. Moreover, both studies developed only heuristic methods, which cannot guarantee to produce optimal or near optimal solutions. In particular, Demir et al. [42] derived a stochastic programming model that necessitates knowing the complete distribution information in advance for uncertain travel times. They solved the model by a sample average approximation method, and applied the method only on small-sized instances. Hrušovskỳ et al. [68] utilized a hybrid simulation-optimization approach that combines deterministic optimization and simulation models to solve their problem. This hybrid simulation-optimization approach iteratively utilizes the deterministic optimization model to obtain deterministic transportation plans, and evaluates the feasibility of the transportation plans under different travel time scenarios by a simulation model. It can thus be concluded that there is no existing study on the development of a robust optimization model and an exact solution method for the CTSNDP or CTSNDP-HC under travel time uncertainty, with delay propagations caused by both transportation and consolidation taken into account.

1.3 Summary of Contributions

This dissertation consists of two studies on the continuous-time service network design problem (CTSNDP). The first study, presented in Chapter 2, investigates the deterministic CTSNDP with holding costs incorporated, which is referred to as the CTSNDP-HC, while the second study, presented in Chapter 3, investigates the robust CTSNDP-HC with uncertain travel times.

As revealed in the literature review, existing studies on the deterministic CTSNDP often overlook the importance of holding costs. The existing discretization methods developed for the SNDP are not effective in solving the CTSNDP, and the existing exact solution methods developed for the CTSNDP without holding costs cannot be applied in solving optimum solutions for the CTNSDP with holding costs. To tackle such research gaps, in the first study of this dissertation we present the first exact solution method for the CTSNDP with holding costs taken into account. Specifically, we develop a novel dynamic discretization discovery (DDD) algorithm for the CTSNDP-HC, which extends the DDD framework to the case with non-zero holding costs. Our distinct contributions in this study are detailed as follows:

- We prove the existence of the finite fully discretized time-expanded network and hence the existence of the finite complete TI formulation for the CTSNDP-HC.

- We propose a new relaxation of the complete TI model for the CTSNDP-HC to provide the valid lower bound solution for the original CTSNDP-HC.

- We adapt the DDD algorithm with the new relaxation method, a customized upper bound heuristic method and a new refinement process to solve the CTSNDP-HC, and also prove that the proposed DDD algorithm can eventually converge to an optimal solution to the CTSNDP-HC.

- We conduct extensive computational experiments to demonstrate the efficiency and effectiveness of the proposed DDD algorithm for the CTSNDP-HC and the benefits that can be gained by taking into account holding costs. The results also reveal that the significance of the benefits turns out to depend upon the connectivity of the underlying physical network and the flexibility of the shipments' time requirements.

While the first study of this dissertation focuses on the CTSNDP-HC with deterministic problem parameters, the second study of this dissertation incorporates travel time uncertainty in the CTSNDP-HC. It aims to develop efficient solution methods that can generate solutions with robust performance against possible changes in travel times. The contributions of the second study are summarized as follows:

- We derive a new formulation of the deterministic CTSNDP-HC that enables us to develop a novel two-stage robust optimization formulation of the CTSNDP-HC under travel time uncertainty. These new formulations are based on the physical network and are defined by decision variables and constraints with indices associated with shipment consolidations.

- Our newly developed two-stage robust optimization model uses a budgeted uncertainty set to incorporate uncertainties in travel times. To our knowledge, it is the first robust optimization model for the SNDP under travel time uncertainty and a continuous-time planning horizon, with non-zero holding costs and consolidation delay propagation taken into account.

- We develop a column-and-constraint generation method as a basic solution method to solve the two-stage robust optimization model, and then enhance it by parameterization and by dynamic parameter adjustment with several novel optimization techniques.

- We conduct extensive computational experiments to evaluate the efficiency and effectiveness of the newly proposed robust optimization model and solution methods, as well as to demonstrate the benefits of incorporating travel time uncertainty through robust optimization in solving the CTSNDP-HC.

This dissertation enriches both deterministic and robust optimization techniques for the continuous-time service network design problem. It not only is of great practical value, but it also lays down a solid foundation for future study on solving various transportation network design problems with holding costs and uncertain travel times.

This dissertation is mainly based on the following working papers:

- Shu, S., Xu, Z. and Baldacci, R. (2022), Incorporating holding costs in continuous-time service network design: new model, relaxation, and exact algorithm.

- Shu, S. and Xu, Z. (2022), Robust continuous-time service network design problem under travel time uncertainty.

In addition, during her PhD study, the candidate has also published the following research works, which do not directly contribute to this thesis:

- Lee, C.-Y., Shu S. and Xu, Z. (2020), Optimal global liner service procurement by utilizing liner service schedules, *Production and Operations Management* **30**(3), 703-714.

- Shen, H., Shu, S., Qin, H. and Wu, Q. (2020), An exact algorithm for the multi-period inspector scheduling problem, *Computers & Industrial Engineering* **145**, 106515.

Chapter 2

Deterministic Continuous-Time Service Network Design with Holding Costs

2.1 Introduction

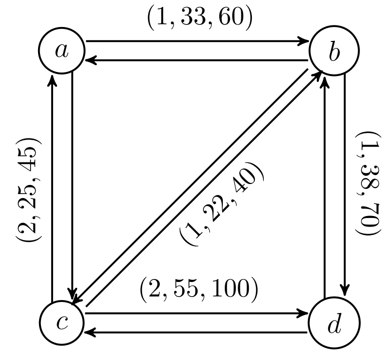
Service network design problems [33] are common and important problems in transportation, telecommunications, logistics, and production–distribution systems. In the freight transportation industry, the less-than-truckload (LTL) motor carriers are typical examples of such systems, where an intensive use of freight consolidation operations are performed to save on transportation costs [109].

The service network design problem considered in this chapter, referred to as the SNDP, can be described as follows. A network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ is given with terminal or node set $\mathcal{N}$ and arc set $\mathcal{A}$. Let $\mathcal{K}$ be a set of commodities, each commodity $k \in \mathcal{K}$ has an origin $o^k \in \mathcal{N}$, a destination $d^k \in \mathcal{N}$, and a transportation demand $q^k \in \mathbb{N}_{>0}$ that must be delivered to the destination from the origin. While flowing along an arc (i, j), a commodity consumes some of the arc capacity; the capacity is obtained by installing on some of the arcs any number of *links*. In network $\mathcal{D}$, also referred as a *flat* network, each arc $(i, j) \in \mathcal{A}$ is associated with the following four attributes: (i) a travel time

$\tau_{ij} \in \mathbb{N}_{>0}$; (ii) a per-unit-of-flow cost $c_{ij}^k \in \mathbb{R}_{>0}$ for each commodity $k \in \mathcal{K}$; (iii) a fixed cost $f_{ij} \in \mathbb{R}_{>0}$; and (iv) a capacity $u_{ij} \in \mathbb{N}_{>0}$. Installing one link on arc (i, j) provides a capacity u_{ij} at a cost f_{ij}. With each commodity $k \in \mathcal{K}$ is also associated an earliest available time $e^k \in \mathbb{N}$ for its departure from the origin o^k, and a due time $l^k \in \mathbb{N}_{>0}$ for its arrival at the destination d^k. We consider the *unsplittable* (or unbifurcated) variant of the problem, where the flow of each commodity is required to follow one route between the origin and the destination, as also considered by Boland et al. [20] and Marshall et al. [81].

The SNDP consists of minimizing the sum of all costs (both fixed and flow costs), while at the same time satisfying demand requirements, as well as capacity and time constraints. The SNDP is known to be strongly $\mathcal{NP}$–hard [56], and various extensions of the SNDP have been studied in the transportation and telecommunications fields [54, 52].

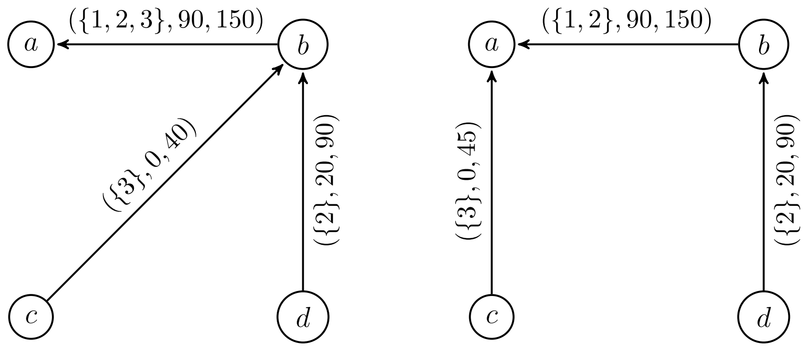
In the SNDP, the decisions are made as to the schedule of the services, this schedule specifying timing information for each possible occurrence of a service during a given time period, such as, the departure and arrival times at the origins, intermediate stops, and destinations. A common technique adopted in the literature for modeling the temporal component is *discretization* [72, 2, 44, 37], where the planning horizon is discretized, and the problem is modeled on a *time-expanded network*. In the network, nodes represent locations in time and space, while arcs or links represent either physical movements between locations or just movements in time at one location. More precisely, the arcs on the network are classified into *dispatch* or *service* arcs and *holding* arcs. A service arc corresponds to the transportation between two locations, and the difference between the periods of these locations is the time elapsed during the transportation activity, whereas a holding arc is directed from one period to another for the same location and represents only time-wise movement. The granularity of the time discretization has an impact on both the computational tractability and the quality of the solutions obtained, and studies have been presented that accurately capture the consolidation opportunities as a Continuous-Time SNDP (CTSNDP) [20, 81].



(a) Network $\mathcal{D}$: (c, f, τ) on arcs

k	o^k	d^k	q^k	e^k	l^k
1	b	a	25	0	160
2	d	a	30	20	180
3	c	a	40	0	180

(b) Instance data



(c) Without holding costs

(d) With holding costs

Figure 2.1 Examples of SNDP solutions

2.1.1 The SNDP with Holding Costs

For many practical applications of the SNDP, *holding costs* have a significant impact on the service and consolidation decisions [98, 25, 88, 70]. These costs are also called *consolidation penalty costs* [100], and can be facility-specific and/or commodity-specific [70, 99, 88]. In the literature, holding costs are classified as *in-transit* and *in-storage*, the in-transit holding cost usually being lower than the in-storage cost. In the context of time-expanded networks, these costs are generally modeled by properly defining the costs associated with the service and holding arcs of the network.

Motivated by the importance of the continuous variant of the SNDP and of the holding costs, in this study we consider the CTSNDP with both *in-transit* and *in-storage* holding costs or simply *holding costs* (CTSNDP-HC). In the following, *in-transit* holding costs are modeled by means of costs c_{ij}^k, whereas to model *in-storage* holding costs, we associate with each commodity $k \in \mathcal{K}$ and node $i \in \mathcal{N}$ a per-unit-of-demand-and-time (holding) cost $h_i^k \in \mathbb{R}_{\geq 0}$. Hereafter, we also use the term *holding costs* to refer to both the *in-transit* and *in-storage* holding costs.

To highlight the importance of considering the holding costs in the SNDP, Figure 2.1 gives a simple example of a SNDP instance with a consideration of holding costs. The example involves three commodities, and Figure 2.1-(a) depicts the underlying network $\mathcal{D}$ where relevant flow and fixed costs and travel times are reported close to each arc. In the example, arcs (i, j) and (j, i) share the same data, and the per-unit-of-flow cost of arc (i, j) is the same for all three commodities. Figure 2.1-(b) gives the different parameters associated with the three commodities. In addition, each arc capacity is assumed to be greater than the total demand of the commodities, i.e., to be 100, and the in-storage per-unit-of-demand-and-time holding cost for each commodity at the different nodes is equal to 0.01.

Figure 2.1-(c) illustrates the optimal solution for the SNDP by disregarding the holding costs. The figure reports on each arc (i, j) the set of commodities consolidated, the departure time from node i, and the arrival time to node j, represented by a triplet $(\{commodities\}, dep.time, arr.time)$. The solution shown in Figure 2.1-(c) shows flow and fixed costs equal to 165 (=1 × 40 (c, b) + 1 × 30 (d, b) + 1 × (25+39+40) (b, a)) and 93 (=22 (c, b) + 38 (d, b) + 33 (b, a)), respectively. The three commodities are consolidated on arc (b, a), where commodities 1 and 3 wait 90 and 50 time units before being consolidated with commodity 2, which arrives at node b at time 90. The total holding cost is therefore equal to 42.5 (=$90 \times 25 \times 0.01 + 50 \times 40 \times 0.01$), thus resulting in a total solution cost equal to 300.5. The solution shows that the average of the per-unit-of-demand-and-time flow costs over arcs (c, d), (d, b) and (b, a) is equal to about 0.02, thus being about twice the in-storage per-unit-of-demand-and-time holding cost of the nodes.

Figure 2.1-(d) illustrates the optimal solution for the SNDP with consideration of the holding costs. Solution 2.1-(d) shows flow and fixed costs equal to 165 (=1 × 40 (c, a) + 1 × 30 (d, b) + 1 × (25+39+40) (b, a)) and 96 (=25 (c, a) + 38 (d, b) +

33 (b,a)), respectively. The holding cost at node b is reduced from 42.5 to 22.5, since commodity 3 is routed on the alternative path (c,a). The total cost of the solution is 283.5, being a cost saving equal to about 6% with respect to the total cost of solution 2.1-(c). Moreover, as shown in the example below, such a cost saving can be arbitrarily large in the best situation.

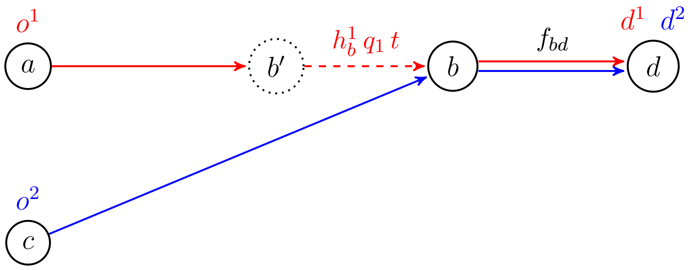


Figure 2.2 Best-case analysis of cost saving for incorporating holding costs

Figure 2.2 shows an example involving four terminals, with two commodities in $\mathcal{K} = \{1,2\}$ having their origins $o^1 = a$ and $o^2 = c$ and their destinations $d^1 = d^2 = d$. In the example, we assume that the commodities' earliest available times e^1 and e^2 are such that commodity 1 of demand q_1 can potentially wait t units of times to be consolidated with commodity 2 at terminal b at an additional holding cost $h_b^1 q_1 t$, modeled by the holding arc (b',b).

If the holding cost at terminal b is ignored when determining the optimal solution, the cost of the last leg $b-d$ excluding the flow cost is equal to $(h_b^1 q_1 t + f_{bd})$ (i.e., the sum of the holding cost and fixed cost), where commodity 1 is consolidated at terminal b with commodity 2 after having waited for t units of time, and the two commodities are then routed together to terminal d. If the holding cost at terminal b is considered and $h_b^1 q_1 t > f_{bd}$, i.e., waiting at terminal b for commodity 1 incurs a holding cost greater than the fixed cost associated with the final arc (b,d), then in the corresponding optimal solution the two commodities are routed on two separate paths, namely (a,b,d) and (c,b,d). As a result, the cost of the last leg $b-d$ excluding the flow cost is now equal to $2f_{bd}$. Since the flow costs are unchanged, a total saving of

$(h_b^1 q_1 t - f_{bd})$ is achieved. This cost saving, with respect to the total cost of the optimal solution, can be arbitrarily large by increasing $h_b^1 q_1 t$ and fixing f_{bd}.

2.1.2 Discretized versus Continuous-Time Models

A time-expanded network provides a useful way of modeling the SNDP, but the corresponding time-index (TI) model (see, for example, [49, 59]) requires a discretization of time known to be fine enough to provide a correct model for the continuous time, i.e., to show that its optimal solution cost is *continuous-time* optimal. What is more, selecting a proper time discretization for the time-expanded network can be challenging. On the one hand, a fine discretization for a time-expanded network can provide good approximations to the original continuous-time problem, but results in a large and often intractable TI model. On the other hand, a coarse discretization is more computationally tractable, but at the expense of a significant loss of solution quality, which is referred to as the *price of discretization* [21].

In this context, it is beneficial to investigate *complete* TI models based on a *complete* discretization of time, i.e., a discretization of time known to be fine enough to provide a correct model for the continuous time. As shown by Boland and Savelsbergh [24], the existence of a complete TI model is not straightforward.

2.1.3 Outlines

In this study, we describe an exact method for the CTSNDP-HC. The method is based on the Dynamic Discretization Discovery (DDD) solution framework proposed by Boland et al. [20] to solve the CTSNDP. The DDD uses successive approximations of a TI model in order to solve the complete TI model, and its correctness relies on the existence of a complete TI model for the problem. The main ingredients of a DDD are:

- A valid *relaxation* of a complete TI model based on a partial discretization.
- A *primal* heuristic that uses the solution provided by the relaxation to compute a valid upper bound on the optimal value of the complete TI model. If the cost

of the primal solution is equal to the solution cost of the relaxation, then the primal solution is proved to be optimal.

- A *refine strategy* that, given a solution of the relaxation, refines the current partial discretization so that the current solution of the relaxation is no longer feasible for the corresponding relaxation model.

The DDD algorithm proposed by Boland et al. [20] for the CTSNDP strongly relies on the assumption that freight can be held at a location at no cost, i.e., in-storage holding costs are equal to zero, and cannot be used to solve the CTSNDP-HC. Our distinct contributions in this study are as follows:

- We prove the existence of a complete TI model for the CTSNDP-HC.
- We derive a new relaxation of the complete TI model based on a mixed-integer linear programming (MIP) model.
- Based on the complete TI model and its new relaxation, we develop a new DDD algorithm with a new upper bound heuristic and a new refinement strategy to solve the CTSNDP-HC.
- We validate the efficiency and effectiveness of the new algorithm via extensive computational experiments and the benefits that can be gained by incorporating holding costs

The remainder of this chapter is organized as follows. We prove the existence of a complete TI model for the CTSNDP-HC in Section 2.2, followed by an detail introduction of the DDD algorithm for solving the CTSNDP-HC in Section 2.3. We report and analyze the results of extensive computational experiments in Section 2.4. Finally, we summary the study in Section 2.5.

2.2 Modeling the CTSNDP-HC on a Finite Time-Expanded Network

Like many other flow-over-time problems (see, for example, [49] and [94]) and network design problems (see, for example, [4], [3], and [86]), the CTSNDP-HC can be approximated by a time-index formulation (TI model) based on a time-expanded network. In this section, we first represent feasible CTSNDP-HC solutions over a time-expanded network, then describe a discretized TI model for the CTSNDP-HC and show the existence of a complete TI model.

2.2.1 Representing Feasible Solutions

A *path* $P^k = (\nu_1^k, \nu_2^k, ..., \nu_{m^k+1}^k)$ for a commodity $k \in \mathcal{K}$ is a path in $\mathcal{D}$ starting from node $\nu_1^k = o^k$ and ending at node $\nu_{m^k+1}^k = d^k$. Associated with path P^k is also the sequence $(a_1^k, a_2^k, ..., a_{m^k}^k)$ of arcs traversed by the path such that $a_n^k = (\nu_n^k, \nu_{n+1}^k) \in \mathcal{A}$ for $n = 1, 2, ..., m^k$; in the following, the two representations of path P^k are used interchangeably. Given a set of departures times $t^k = (t_1^k, t_2^k, ..., t_{m^k}^k)$ associated with the nodes of the path, path P^k is k-feasible, and we denote it with the pair $\mathcal{W}^k = (P^k, t^k)$, if values t_n^k, $n = 1, \ldots, m^k$, satisfy the following system of inequalities:

$$t_n^k \geq e^k, \qquad n = 1, \tag{2.1a}$$

$$t_n^k \geq t_{n-1}^k + \tau_{a_{n-1}^k}, \qquad n = 2, \ldots, m^k, \tag{2.1b}$$

$$t_n^k + \tau_{a_n^k} \leq l^k, \qquad n = m^k, \tag{2.1c}$$

Inequalities (2.1a) and (2.1c) impose that the departure and arrival times at the origin and destination are within the required time limits e^k and l^k, respectively, where the term $t_n^k + \tau_{a_n^k}$ coincides also with the departure time at the destination d^k. Inequalities (2.1b) impose feasible departure times at the intermediate nodes of the path.

We also associate with each arc $a_n^k \in P^k$, $n = 1, 2, \ldots, m^k$, a departure time that corresponds to time t_n^k. A *feasible* solution $\mathcal{W} = \{\mathcal{W}^k\}_{k \in \mathcal{K}}$ of the CTSNDP-HC is a collection of $|\mathcal{K}|$ paths, one feasible path for each commodity. We assume that for each commodity $k \in \mathcal{K}$, the difference $(l^k - e^k)$ of 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 CTSNDP-HC.

Given a k-feasible timed path $\mathcal{W}^k$, the *holding plan* of the path is defined as the set of the waiting times H_n^k, $n = 1 \ldots, m^k + 1$, at the different nodes of the path that can be computed as follows:

$$H_n^k = \begin{cases} t_n^k - e^k, & n = 1, \\ t_n^k - (t_{n-1}^k + \tau_{a_{n-1}^k}), & n = 2, \ldots, m^k, \\ l^k - (t_{n-1}^k + \tau_{a_{n-1}^k}), & n = m^k + 1. \end{cases} \tag{2.2}$$

Associated with a CTSNDP-HC solution $\mathcal{W}$ is the consolidation plan, where each element defines that a subset of commodities are transported together through an arc of the solution. More precisely, we denote with $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|\mathcal{C}|}\}$ a consolidation plan. Each $\mathcal{C}_r = (\alpha_r, J_r)$, $r = 1, 2, \ldots, |\mathcal{C}|$, denotes a consolidation on arc $\alpha_r \in \mathcal{A}$, with J_r being a set of pairs (k, n) with $a_n^k = \alpha_r$, indicating that such commodities k are shipped together through arc α_r when they are routed through the n-th arcs of their paths P^k in solution $\mathcal{W}$.

For each arc $\alpha = (i, j) \in \cup_{k \in \mathcal{K}} P^k$, we define $\Theta(\alpha) = \{t_n^k : k \in \mathcal{K}, a_n^k \in P^k, a_n^k = \alpha\}$ as the set of departure times associated with the arcs of paths in solution $\mathcal{W}$. Accordingly, for each of such departure times $t \in \Theta(\alpha)$, a consolidation $(\alpha, I(\alpha, t))$ can be defined for arc α, where $I(\alpha, t)$ defined below indicates the set of commodities that are shipped together through α with departure time t in solution $\mathcal{W}$:

$$I(\alpha, t) = \{(k, n) : k \in \mathcal{K}, a_n^k = \alpha \in P^k \text{ and } t_n^k = t\}.$$

Table 2.1 The consolidation plan for the example of Figure 2.1-(c)

k	P^k	$\mathcal{C}$	
		α	J
1	(b,a)	(b,a)	$\{(1,1),(2,2),(3,2)\}$
2	(d,b,a)	(d,b)	$\{(2,1)\}$
3	(c,b,a)	(c,b)	$\{(3,1)\}$

With this, a consolidation plan $\mathcal{C}$ can be defined by solution $\mathcal{W}$ as follows:

$$\mathcal{C} = \{(\alpha, I(\alpha,t)) : \forall \alpha \in \cup_{k\in\mathcal{K}} P^k, t \in \Theta(\alpha), I(\alpha,t) \neq \emptyset\}.$$

The cost $z(\mathcal{W})$ of a CTSNDP-HC solution $\mathcal{W}$ can then be computed as a function of the holding and consolidation plans:

$$z(\mathcal{W}) = \sum_{\mathcal{C}_r \in \mathcal{C}} f_{\alpha_r} \left\lceil \frac{\sum_{(k,n)\in J_r} q^k}{u_{\alpha_r}} \right\rceil + \sum_{k\in\mathcal{K}} \sum_{n=1}^{m^k} c^k_{a^k_n} q^k + \sum_{k\in\mathcal{K}} \sum_{n=1}^{m^k+1} h^k_{\nu^k_n} q^k H^k_n, \tag{2.3}$$

where the three terms represent the fixed, flow and holding costs, respectively.

Alternatively, a feasible CTSNDP-HC solution can also be defined by (i) a routing plan $\mathcal{P} = \{P^k\}_{k\in\mathcal{K}}$, (ii) a consolidation plan $\mathcal{C}$ and (iii) a set of departure times $\{t^k\}_{k\in\mathcal{K}}$, such that for each $k \in \mathcal{K}$, path P^k is k-feasible, and that from the consolidation plan $\mathcal{C}$, the waiting times $\{H^k\}_{k\in\mathcal{K}}$ can be computed by expressions (2.2). We define $\mathcal{S} = (\mathcal{P}, \mathcal{C})$ as a *flat solution*, and the flat solution $\mathcal{S}$ is *implementable* if a set of departure times $\{t^k\}_{k\in\mathcal{K}}$ satisfying condition (iii) exists.

As an example, Table 2.1 gives the consolidation plan $\mathcal{C}$ associated with the example of Figure 2.1-(c). For each of the three commodities, the table gives the corresponding path P^k. The consolidation plan $\mathcal{C}$ shows that there are three consolidations associated with arcs (b,a), (d,b), and (c,b). In particular, all three commodities are consolidated on arc $\alpha = (b,a)$.

2.2.2 A Time-Index Formulation for the CTSNDP-HC

We consider a time-expanded network with a discretization level Δ, $\mathcal{D}_{\mathcal{T}}^{\Delta} = (\mathcal{N}_{\mathcal{T}}^{\Delta}, \mathcal{H}_{\mathcal{T}}^{\Delta} \cup \mathcal{A}_{\mathcal{T}}^{\Delta})$ where $\mathcal{T} = (\mathcal{T}_i)_{i \in \mathcal{N}}$ is a set of time points with $\mathcal{T}_i = \{0, \Delta, 2\Delta, ..., M\Delta\}$ for all $i \in \mathcal{N}$ and for $M \in \mathbb{N}_{>0}$ with $M = max_{k \in \mathcal{K}}\{\lceil l^k/\Delta \rceil\}$. The node set $\mathcal{N}_{\mathcal{T}}^{\Delta}$ consists of each time node (i,t) associated with each $i \in \mathcal{N}$ and $t \in \mathcal{T}_i$. The set of arcs of $\mathcal{D}_{\mathcal{T}}^{\Delta}$ contains two subsets of arcs:

- *Holding* arcs $\mathcal{H}_{\mathcal{T}}^{\Delta}$. For every node $i \in \mathcal{N}$, and every $t \in \mathcal{T}_i \setminus \{M\Delta\}$, there is an arc $((i,t),(i,t+\Delta))$ representing a holding time of Δ time units at node i.
- *Dispatch* or *service* arcs $\mathcal{A}_{\mathcal{T}}^{\Delta}$. For every arc $(i,j) \in \mathcal{A}$, and every node $(i,t) \in \mathcal{N}_{\mathcal{T}}^{\Delta}$, there is an arc $((i,t),(j,\bar{t}))$ with $i \neq j$ representing a dispatch from node i at time t arriving at time $\bar{t}$ at node j with $\bar{t} = t + \Delta\lceil \tau_{ij}/\Delta \rceil$ and $\bar{t} \leq M\Delta$, and since $\tau_{ij} \leq \Delta\lceil \tau_{ij}/\Delta \rceil$, the condition $\bar{t} \geq t + \tau_{ij}$ holds, thus guaranteeing that the feasible solutions of a TI formulation based on graph $\mathcal{D}_{\mathcal{T}}^{\Delta}$ (see below) are also feasible for the CTSNDP-HC.

Network $\mathcal{D}_{\mathcal{T}}^{\Delta}$ is also known in the literature as a *condensed* time-expanded network. Below, we model the CTSNDP-HC using a TI formulation based on graph $\mathcal{D}_{\mathcal{T}}^{\Delta}$, denoted as SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$).

Let $y_{ij}^{t\bar{t}}$ be a nonnegative integer variable representing the number of times that arc $(i,j) \in \mathcal{A}$ is used to serve the dispatches from node i at time t arriving at time $\bar{t}$ in j, and let $x_{ij}^{kt\bar{t}}$ be 0-1 variable equal to 1 if commodity $k \in \mathcal{K}$ is routed along arc $(i,j) \in \mathcal{A}$ departing from i at time t and arriving at j at time $\bar{t}$, 0 otherwise. Moreover, let w_i^k be a nonnegative variable denoting the holding or waiting time of commodity k at node i. Formulation SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$) is as follows:

$$z(\mathcal{D}_{\mathcal{T}}^{\Delta}) = \min \sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}} f_{ij} \cdot y_{ij}^{t\bar{t}} + \sum_{k \in \mathcal{K}} \sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}} (c_{ij}^k q^k) \cdot x_{ij}^{kt\bar{t}} + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} (h_i^k q^k) \cdot w_i^k \tag{2.4}$$

$$s.t. \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}\cup\mathcal{H}_{\mathcal{T}}^{\Delta}} x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}^{\Delta}\cup\mathcal{H}_{\mathcal{T}}^{\Delta}} x_{ji}^{k\bar{t}t} = \begin{cases} 1 & (i,t) = (o^k, e^k), \\ -1 & (i,t) = (d^k, l^k), \\ 0 & \text{otherwise}, \end{cases}$$

$$\forall k \in \mathcal{K}, (i,t) \in \mathcal{N}_{\mathcal{T}}^{\Delta}, \quad (2.5)$$

$$\sum_{k\in\mathcal{K}} q^k x_{ij}^{kt\bar{t}} \leq u_{ij} y_{ij}^{t\bar{t}}, \quad \forall\ ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}, \quad (2.6)$$

$$w_i^k = \begin{cases} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ji}^{k\bar{t}t}, & \text{otherwise}, \end{cases} \quad \forall\ i \in \mathcal{N}, \quad \forall\ k \in \mathcal{K},$$

$$(2.7)$$

$$x_{ij}^{kt\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta} \cup \mathcal{H}_{\mathcal{T}}^{\Delta}, k \in \mathcal{K}, \quad (2.8)$$

$$y_{ij}^{t\bar{t}} \in \mathbb{N}, \quad \forall((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}, \quad (2.9)$$

$$w_i^k \geq 0, \quad \forall i \in \mathcal{N}, k \in \mathcal{K}. \quad (2.10)$$

In the above formulation, the objective function (2.4) aims to minimize the total cost computed as the sum of the fixed, flow and holding costs, respectively. Constraints (2.5) are *flow conservation constraints* ensuring that each commodity $k \in \mathcal{K}$ is routed along a single path starting from its origin after its earliest available time e^k and ending at its destination before its due time l^k. For each commodity $k \in \mathcal{K}$, the timed path starts from time node (o^k,e^k) and ends at time node (d^k,l^k), and holding arcs allow each commodity $k \in \mathcal{K}$ to arrive at its destination earlier than l^k or depart from its origin later than e^k. Constraints (2.6) ensure that the flow on each service arc does not exceed the capacity installed on the arc. Constraints (2.7) define the values of variables w_i^k, computed as the difference between the departure and arrival times. Finally, constraints (2.8), (2.9) and (2.10) state the domains of the decision variables.The above formulation contains as a special case the TI formulation described by Boland et al. [20] and used to solve the CTSNDP. Indeed, when each coefficient h_i^k

is assumed to be equal to zero, the corresponding decision variables w_i^k are no more necessary in the formulation. We denote with SND($\mathcal{D}_{\mathcal{T}}^{\Delta}$) the resulting formulation.

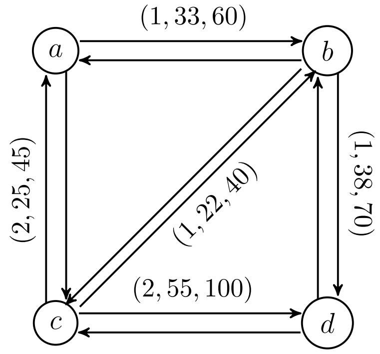
Due to the definition of the time-expanded network $\mathcal{D}_{\mathcal{T}}^{\Delta}$ such that for each arc $(i, j) \in \mathcal{A}$ we have $\Delta\lceil \tau_{ij}/\Delta \rceil \geq \tau_{ij}$, any feasible solution of formulation SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$) is also a feasible solution for the CTSNDP-HC. Nevertheless, an optimal SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$) solution is not necessarily an optimal CTSNDP-HC solution, due to the discretization factor Δ, i.e., $z(\mathcal{D}_{\mathcal{T}}^{\Delta})$ provides a valid upper bound on the optimal CTSNDP-HC solution cost.

It is not straightforward to prove the existence of a value $\hat{\Delta}$ such that $z(\mathcal{D}_{\mathcal{T}}^{\hat{\Delta}})$ provides the optimal solution cost of the CTSNDP-HC. Indeed, as observed by Boland and Savelsbergh [24], for some problems such as the TSPTW, a simple combinatorial argument suffices to show the existence of a complete TI model, whereas for other problems, such as the continuous-time inventory routing problem [75], such $\hat{\Delta}$ may be smaller than one and is difficult to identify.

2.2.3 Existence of a Finite Time-Expanded Network for the CTSNDP-HC

For the CTSNDP, the existence of a complete TI model has been shown by Boland et al. [20], who noted that when the travel times and time window limits are integer-valued, the set of times points e^k, for some commodity $k \in \mathcal{K}$, or of the form $e^k + \sum_{a \in P} \tau_a$, for some path P in G originating at o^k, suffice to compose a complete TI model. The observation is that the dispatch times of a path P can be shifted to be as early as possible without changing any consolidations so that the total cost is not changed, and strictly relies on the assumption that in-storage holding costs are equal to zero. For the CTSNDP-HC, due to the presence of nonzero holding costs in the problem objective, such an observation is no longer valid.

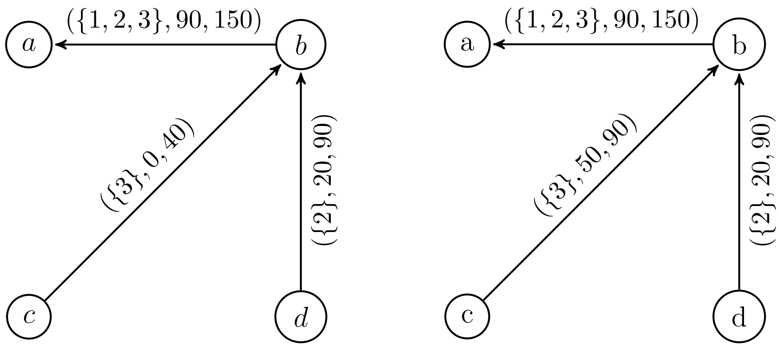
To illustrate the case, Figure 2.3 considers the example of Figure 2.1 where the solution of Figure 2.3-(c) depicts the optimal solution of the CTSNDP. The alternative CTSNDP optimal solution represented by Figure 2.3-(d), where the departure time at



(a) Network $\mathcal{D}$: (c, f, τ) on arcs

k	o^k	d^k	q^k	e^k	l^k
1	b	a	25	0	160
2	b	a	30	20	180
3	c	a	40	0	180

(b) Instance data



(c) An optimal solution for the CTSNDP

(d) Alternative optimal solution for the CTSNDP

Figure 2.3 Examples showing that for the CTSNDP-HC the transformation of Boland et al. [20] cannot be applied

the origin of commodities 3 is equal to 50, can be shifted to be as early as possible, thus obtaining the departure time of the solution of Figure 2.3-(c) without changing the consolidations on arc (a, b). When considering the CTSNDP-HC, if the per-unit-of-demand-and-time holding costs for commodity 3 at terminals b and c are equal to 0.01 and 0.005, respectively, commodity 3 incurs a holding cost equal to $0.01 \times 50 \times 40$ for the solution of Figure 2.3-(c), whereas it incurs a lower holding cost equal to $0.005 \times 50 \times 40$ for the solution of Figure 2.3-(d). Therefore, the time node (c, 50), which is not part of a complete TI model for the CTSNDP, must be considered when solving the CTSNDP-HC.

In order to show that a complete TI model exists for the CTSNDP-HC, it is necessary to prove that, given a flat solution $\mathcal{S}$, a linear programming model (LP) can be defined to determine optimal departure times t^k for each $k \in \mathcal{K}$ that are integers. This argument suffices to show that a complete TI model exists. The LP argument was also used by Boland et al. [23] for a network scheduling problem.

Consider a flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$, with a routing plan $\mathcal{P}$ and consolidation plan $\mathcal{C}$ associated with $\mathcal{P}$. We denote with $z_{fc}(\mathcal{S})$ the cost of the flat solution $\mathcal{S}$, that is, the sum of its fixed and flow costs:

$$z_{fc}(\mathcal{S}) = \sum_{\mathcal{C}_r \in \mathcal{C}} f_{\alpha_r} \left\lceil \frac{\sum_{(k,n)\in J_r} q^k}{u_{\alpha_r}} \right\rceil + \sum_{k\in\mathcal{K}} \sum_{n=1}^{m^k} c^k_{a^k_n} q^k.$$

For each $k \in \mathcal{K}$ and each node $v^k_n \in P^k$, $n = 1, 2, ..., m^k + 1$, we define nonnegative continuous variables $\pi^k_{v_n}$ and $t^k_{v_n}$ as the arrival and departure times of commodity k at node v^k_n, respectively. Moreover, for each $\mathcal{C}_r = (\alpha_r, J_r)$, $r = 1, ..., |\mathcal{C}|$, we define a nonnegative continuous variable $\hat{t}_{\mathcal{C}_r} \geq 0$ representing the consolidation time of the commodities in J_r on arc α_r, i.e., the joint departure time of all commodities $(k, n) \in J_r$.

If the flat solution $\mathcal{S}$ is implementable, then the following LP formulation, denoted as *implementable model* ($IM(\mathcal{S})$), computes corresponding optimal departure times for flat solution $\mathcal{S}$ of cost $z_{fc}(\mathcal{S}) + z_w(\mathcal{S})$:

$$\begin{aligned}
z_w(\mathcal{S}) = \min \quad & \sum_{k\in\mathcal{K}} \sum_{n=1}^{m^k+1} (h^k_{v_n} q^k) \cdot (t^k_{v_n} - \pi^k_{v_n}) && (2.11)\\
s.t. \quad & \pi^k_{v_{n+1}} - t^k_{v_n} = \tau_{v_n v_{n+1}}, \quad \forall\, k \in \mathcal{K},\, n = 1, ..., m^k, && (2.12)\\
& t^k_{v_n} - \pi^k_{v_n} \geq 0, \quad \forall k \in \mathcal{K},\, n = 1, ..., m^k + 1, && (2.13)\\
& \hat{t}_{\mathcal{C}_r} - t^k_{v_n} = 0, \quad \forall (k, n) \in J_r,\, r = 1, ..., |\mathcal{C}|, && (2.14)\\
& \pi^k_{o^k} = e^k, \quad \forall k \in \mathcal{K}, && (2.15)\\
& t^k_{d^k} = l^k, \quad \forall k \in \mathcal{K}, && (2.16)\\
& \pi^k_{v_n} \geq 0, \quad \forall\, k \in \mathcal{K},\, n = 1, ..., m^k + 1, && (2.17)\\
& t^k_{v_n} \geq 0, \quad \forall k \in \mathcal{K},\, n = 1, ..., m^k + 1, && (2.18)\\
& \hat{t}_{\mathcal{C}_r} \geq 0, \quad \forall r = 1, ..., |\mathcal{C}|. && (2.19)
\end{aligned}$$

The objective function (2.11) aims to minimize the total holding cost associated with the flat solution. Constraints (2.12), (2.13), (2.15) and (2.16) define the arrival and departure times according to paths P^k, $k \in \mathcal{K}$, respectively. Constraints (2.14) impose that the departure times at the intermediate nodes follow the consolidation plan $\mathcal{C}$.

The following proposition implies the existence of a complete TI model for the CTSNDP-HC.

Proposition 1. *If e_k, l_k and τ_{ij} are integer-valued and the flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$ is implementable, then for formulation IM($\mathcal{S}$) there is an integral optimal solution.*

Proof. The matrix associated with constraints (2.12), (2.13) and (2.14) has coefficients in $\{0, 1, -1\}$ and each column of its transpose has exactly two nonzero elements of different sign. Hence, it is totally unimodular and together with the bound constraints (2.15)-(2.19) ensure that the extreme points of (2.12)-(2.19) are integral (see, for example, [18]). □

Notice that the above proposition does not suggest a specific value of the discretization Δ. It is easy to see that if ratios τ_{ij}/Δ, e_k/Δ and l_k/Δ are integer-valued for some $\Delta \in \mathbb{N}_{>0}$, $z(\mathcal{D}^{\Delta}_{\mathcal{T}})$ corresponds to the optimal solution cost of the CTSNDP-HC, and that in the worst case we have $\Delta = 1$. In practice, the size of the complete TI model can be computationally intractable, but in the next section we describe an exact algorithm aimed at finding the optimal CTSNDP-HC solution by solving a set of reduced models of the complete TI model.

2.3 Dynamic Discritization Discovery Algorithm for the CTSNDP-HC

In this section, we introduce a dynamic discritization discovery (DDD) algorithm for the CTSNDP-HC. We start by an overview of the scheme of DDD algorithm in Section 2.3.1. Then we describe in detail the different components of the DDD algorithm for the CTSNDP-HC. We illustrate how we construct and strengthen the relaxation of the complete TI model for the CTSNDP-HC in Section 2.3.2 and Section 2.3.3, followed by how the partially time-expanded network is initialized in Section 2.3.4 and the details of the algorithm used to compute a valid upper bound in Section 2.3.5. We present the refinement strategy in Section 2.3.6. Moreover, we shows that our proposed DDD algorithm solves the CTSNDP-HC to optimality after a finite number of iterations in Section 2.3.7.

2.3.1 An Overview of the Dynamic Discritization Discovery Algorithm

A complete TI model for the CTSNDP-HC implies the existence of a discretization $\hat{\Delta}$ and of the corresponding fully time-expanded network $\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$ such that the optimal solution cost $z(\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}})$ of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$) is the optimal solution cost of the CTSNDP-HC. However, the size of the network $\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$ can be prohibitively large, and the resulting TI model impractical to be solved by conventional techniques.

For the CTSNDP, Boland et al. [20] proposed a DDD algorithm that dynamically and iteratively determines the time points that are present in an optimal solution. Let $\mathcal{D}_{\mathcal{T}} = (\mathcal{N}_{\mathcal{T}}, \mathcal{H}_{\mathcal{T}} \cup \mathcal{A}_{\mathcal{T}})$ be a partially time-expanded network that contains only a small subset of the time points of the fully time-expanded network, i.e., $|\mathcal{N}_{\mathcal{T}}| \ll |\mathcal{N}_{\hat{\mathcal{T}}}^{\hat{\Delta}}|$. As illustrated in Algorithm 1, the DDD algorithm starts by properly initializing the partially time-expanded network $\mathcal{D}_{\mathcal{T}}$, and at each iteration of the algorithm, a relaxation model SND($\mathcal{D}_{\mathcal{T}}$) is solved to compute a valid lower bound LB on the CTSNDP. An upper bound is also computed using the lower bound solution and the algorithm iterates until a predefined optimality tolerance is reached. The partially time-expanded network $\mathcal{D}_{\mathcal{T}}$ is initialed and modified (or *refined*) whenever the optimality tolerance is not reached to ensure the computation of a valid lower bound. The construction of the network $\mathcal{D}_{\mathcal{T}}$, together with the relaxation model SND($\mathcal{D}_{\mathcal{T}}$), ensures that whenever the lower bound solution is not proved to be an optimal CTSNDP solution, the network contains at least one arc, say arc $((i,t),(j,t'))$ such that $t' < t + \tau_{ij}$, i.e., the length of the arc is *too short* and arc $((i,t),(j,t'))$ is a *short-arc*. The network is therefore

Algorithm 1: DDD algorithm for the CTSNDP

Input: CTSNDP defined on a flat network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$
Output: Solution $\mathcal{W} = \{\mathcal{W}^k\}_{k \in \mathcal{K}}$ of cost UB
begin
  `// Initialization`
1   $UB \leftarrow +\infty$, $LB \leftarrow -\infty$, $gap \leftarrow +\infty$, $\mathcal{W} \leftarrow \emptyset$;
  `// Initialize the partially time-expanded network`
2   $\mathcal{D}_\mathcal{T} \leftarrow (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T})$;
  `// Termination condition`
3   **while** *gap* > *optimality tolerance* **do**
    `// Solution of the relaxation`
4   Solve SND($\mathcal{D}_\mathcal{T}$) and set LB equal to the optimal solution cost of SND($\mathcal{D}_\mathcal{T}$);
    `// Compute a feasible CTSNDP solution`
5   Compute a valid upper bound z based on the solution defined by the relaxation;
6   **if** $z < UB$ **then**
7       $UB \leftarrow z$;
8   Update solution $\mathcal{W}$;
9   **end**
    `// Compute the optimality tolerance`
10     $gap \leftarrow (UB - LB)/UB$;
    `// Check the optimality condition`
11   **if** *gap* > *optimality tolerance* **then**
      `// Optimality not reached`
12   Based on the solution of SND($\mathcal{D}_\mathcal{T}$), refine the network $\mathcal{D}_\mathcal{T}$ to correct the length of at least one *short arc*;
13   **end**
14   **end**
15   **return** Solution $\mathcal{W}$ of cost UB;
16 **end**

refined by adding new time points and modifying corresponding arcs, correcting the length of short-arc $((i, t), (j, t'))$ to be its actual value. The correctness of the algorithm follows on from the validity of bounds LB and UB. The convergence of the method relies on the refinement strategies, which guarantee that the final relaxation model will eventually converge to the complete TI model.

In this study, we adapt the DDD Algorithm 1 in a novel way to solve the more general CTSNDP-HC as summarized below.

(1) At Step 4, we solve a novel relaxation of formulation SND-HC($\mathcal{D}_\mathcal{T}^{\hat{\Delta}}$). The relaxation relies on both the definition of the network $\mathcal{D}_\mathcal{T}$ and on a formulation obtained by relaxing equations (2.7) defining the holding variables $\boldsymbol{w}$ (see Section 2.3.2).

(2) At Step 5, we compute a valid upper bound UB based on the flat solution $\mathcal{S}$ defined by the relaxation using a new heuristic algorithm accounting for the holding costs (see Section 2.3.5).

(3) At Step 12, we extend the refinement strategy used by Boland et al. [20] to add new time points based on the definition of variables $\boldsymbol{w}$ (see Section 2.3.6).

In the reminder of this section, we give the details of the DDD algorithm adapted for the CTSNDP-HC.

2.3.2 A Relaxation of the CTSNDP-HC

In this section, we describe a valid relaxation for the CTSNDP-HC. We start by a simple observation that if $h_i^k = 0$, $\forall k \in \mathcal{K}, i \in \mathcal{N}$, then the CTSNDP-HC reduces to the CTSNDP, and any valid lower bound (including the optimal objective value) for the CTSNDP is also a valid lower bound for the CTSNDP-HC. Boland et al. [20] described a lower bound for the CTSNDP based on a special case of formulation SND-HC($\mathcal{D}_{\mathcal{T}}$) where variables w_i^k, for $k \in \mathcal{K}, i \in \mathcal{N}$, and constraints (2.7) are disregarded, since in-storage holding costs h_i^k, $\forall k \in \mathcal{K}, i \in \mathcal{N}$, are all equal to zero. They show that the following three properties of the partially time-expanded network $\mathcal{D}_{\mathcal{T}}$ suffice to provide a valid lower bound on $z(\mathcal{D}_{\mathcal{T}}^{\hat{\Delta}})$.

Property 1. *For all commodities $k \in \mathcal{K}$, the nodes (o^k, e^k) and (d^k, l^k) are in $\mathcal{N}_{\mathcal{T}}$.*

Property 2. *Every arc $((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$ has $\bar{t} \leq t + \tau_{ij}$.*

Property 3. *For every arc $a = (i,j) \in \mathcal{A}$ in the flat network $\mathcal{D}$, and for every node (i,t) in the partially time-expanded network $\mathcal{D}_{\mathcal{T}}$, there is an arc $a' = ((i,t),(j,\bar{t}))$ in $\mathcal{D}_{\mathcal{T}}$ for some $\bar{t} \in \mathcal{N}_{\mathcal{T}}$ (a' is called a timed copy of arc a in $\mathcal{A}_{\mathcal{T}}$).*

Let $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}})$ be an optimal solution of formulation SND-HC($\mathcal{D}_{\mathcal{T}}^{\hat{\Delta}}$) with zero holding costs (i.e., an optimal CTSNDP solution) and let $\overline{\mathcal{A}} = \{((i,t),(j,t+\tau_{ij})) \in \mathcal{A}_{\mathcal{T}}^{\hat{\Delta}} : \overline{y}_{ij}^{t,t+\tau_{ij}} > 0\}$ be the set of arcs traversed by the commodities in the solution. For any arc $a = ((i,t),(j,t+\tau_{ij})) \in \overline{\mathcal{A}}$, define $\rho_i(t) = \text{argmax}\{s \in \mathcal{T}_i : s \leq t\}$. The existence of such $\rho_i(t)$ for each arc $a = ((i,t),(j,t+\tau_{ij})) \in \overline{\mathcal{A}}$ is ensured by the three properties. Indeed, with denoting $\underline{\tau}_{ij}$ as the length of any shortest-time path from node i to node j in the flat network $\mathcal{D}$, for each $k \in \mathcal{K}$ and each $i \in \mathcal{N}$, a time node $(i,t) \in \mathcal{N}_\mathcal{T}$ exists with $t \leq e_k + \underline{\tau}_{o_k\, i}$. Further, by Property 3 a timed-copy arc $((i,\rho_i(t)),(j,t')) \in \mathcal{A}_\mathcal{T}$ of arc a exists in $\mathcal{D}_\mathcal{T}$ for some $(j,t') \in \mathcal{N}_\mathcal{T}$, and define $\sigma(a)$ to be any such t'.

Proposition 2 below shows that formulation SND-HC($\mathcal{D}_\mathcal{T}$) with zero holding costs defined over a network $\mathcal{D}_\mathcal{T}$ satisfying Properties 1, 2 and 3 is a valid relaxation for formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$).

Proposition 2 (**Boland et al. [20]**). *If the in-storage holding costs h_i^k are all equal to zero, the mapping of solution $(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}})$ into a solution $(\boldsymbol{x},\boldsymbol{y})$ of formulation SND-HC($\mathcal{D}^{\Delta}_{\mathcal{T}}$) defined by $\mu : \overline{\mathcal{A}} \to \mathcal{A}_\mathcal{T}$ with $\mu(a) = ((i,\rho_i(t)),(j,\sigma(a)))$ and computed by the following expressions for each $\tilde{a} = ((i,\tilde{t}),(j,\tilde{t}')) \in \mathcal{A}_\mathcal{T}$:*

$$x_{ij}^{k\tilde{t}\tilde{t}'} = \sum_{\substack{a=((i,t),(j,t+\tau_{ij}))\in\overline{\mathcal{A}}:\\ \mu(a)=\tilde{a}}} \overline{x}_{ij}^{kt,t+\tau_{ij}} \quad \textit{and} \quad y_{ij}^{\tilde{t}\tilde{t}'} = \sum_{\substack{a=((i,t),(j,t+\tau_{ij}))\in\overline{\mathcal{A}}:\\ \mu(a)=\tilde{a}}} \overline{y}_{ij}^{t,t+\tau_{ij}}, \tag{2.20}$$

corresponds to a feasible solution of formulation SND-HC($\mathcal{D}_\mathcal{T}$) of the same cost of solution $(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}})$.

The proof of the above proposition is based on the observation that, for each commodity $k \in \mathcal{K}$, to each path $\overline{P}^k = (a_1^k,\ldots,a_{m^k}^k)$, $a_h^k \in \overline{\mathcal{A}}$, $h = 1,\ldots,m^k$, induced by solution $(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}})$ with $a_h^k = ((i_h^k,t_h^k),(i_{h+1}^k,t_h^k+\tau_{i_h^k i_{h+1}^k}))$ and $t_{h+1}^k \geq t_h^k + \tau_{i_h^k i_{h+1}^k}$ for $h = 1,..,m^k-1$, corresponds a feasible path $P^k = (\mu(a_1^k),\ldots,\mu(a_{m^k}^k))$ in $\mathcal{D}_\mathcal{T}$ with appropriate holding arcs.

The above proposition relies on the fact that adding additional holding arcs does not result in additional cost. In the presence of nonzero holding costs, given an optimal $(\overline{\boldsymbol{x}},\overline{\boldsymbol{y}},\overline{\boldsymbol{w}})$ solution of formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$) and corresponding paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$, for each $k \in \mathcal{K}$ we have that:

$$\overline{w}^k_{i^k_h} = \begin{cases} t^k_h - e^k, & h = 1, \\ l^k - (t^k_{h-1} + \tau_{i_{h-1}i_h}), & h = m^k + 1, \\ t^k_h - (t^k_{h-1} + \tau_{i^k_{h-1}i^k_h}), & \text{otherwise,} \end{cases} \quad \forall\ h = 1, \dots, m^k + 1.$$

If we now consider the mapped solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ based on Proposition 2, since for each $k \in \mathcal{K}$ variables $\boldsymbol{w}$ depend only on variables $\boldsymbol{x}$ (see equation (2.7)) and $\rho_{i^k_h}(t^k_h) \leq t^k_h$ and $\sigma(a^k_h) \leq t^k_h + \tau_{i^k_h i^k_{h+1}}$, $\forall\ h = 1, \dots, m^k$, we have:

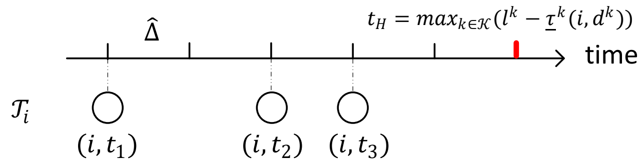
$$w^k_{i^k_h} = \begin{cases} \rho_{i^k_h}(t^k_h) - e^k \leq \overline{w}^k_{i^k_h}, & h = 1, \\ l^k - \sigma(a^k_{h-1}) \geq \overline{w}^k_{i^k_h}, & h = m^k + 1, \\ \rho_{i^k_h}(t^k_h) - \sigma(a^k_{h-1}) \geq \overline{w}^k_{i^k_h} \text{ or } \rho_{i^k_h}(t^k_h) - \sigma(a^k_{h-1}) \leq \overline{w}^k_{i^k_h}, & \text{otherwise,} \end{cases} \quad \forall\ h = 1, \dots, m^k + 1.$$

Therefore, the total holding cost associated with solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ is not proved to be less than or equal to the total holding cost associated with solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ and even under Properties 1-3, formulation SND-HC($\mathcal{D}_{\mathcal{T}}$) is not a valid relaxation for formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$).

To obtain a valid relaxation for formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$), we derive a relaxation of equations (2.7) based on the following observations. Let $P^k = (a^k_1, \dots, a^k_{m^k})$ with $a^k_h = ((i^k_h, t^k_h), (i^k_{h+1}, t^k_{h+1}))$, $h = 1, \dots, m^k$, $k \in \mathcal{K}$, be a path in network $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ representing a feasible k-path, where we denote with $\bar{t}^k_h$ and t^k_h the arrival and departure times at node i^k_h, $h = 1, .., m^k + 1$, respectively, with $\bar{t}^k_1 = e^k$:

(i) On the partially time-expanded network $\mathcal{D}^{\Delta}_{\mathcal{T}}$ satisfying Properties 1-3, to each arrival time $\bar{t}^k_h$, $h = 2, .., m^k+1$, we can associate a lower bound $\check{t} \leq \bar{t}^k_h$ computed as $\check{t} = \sigma(a^k_{h-1})$. In addition, we associate an upper bound $\rho_{i^k_h}(t^k_h) \leq t^k_h \leq \hat{t} = \xi_{i^k_h}(t^k_h)$ with the departure time t^k_h, thus obtaining an upper bound on the holding time at node i_h, i.e., $t^k_h - \bar{t}^k_h \leq \hat{t} - \check{t}$.

(ii) Let $T(P^k)$ be the total transit time of path P^k, computed as $T(P^k) = \sum_{h=1}^{m^k} \tau_{a^k_h}$. Then, the total holding time of path P^k must be equal to $l^k - e^k - T(P^k)$ since



(a) Time points in $\mathcal{T}_i$

t	$\vec{t_i}(t)$	$\xi_i(t)$
t_1	t_2	t_2
t_2	t_3	t_2
t_3	t_H	t_H

(b) Value of function $\xi_i(t)$

Figure 2.4 Illustration of expression (2.22)

each commodity $k \in \mathcal{K}$ leaves its origin o^k at time e^k and arrives at its destination d^k at time l^k.

For a commodity $k \in \mathcal{K}$, let $\underline{\tau}^k(i, j)$ denote the time of the shortest-time path from node i to node j in the flat network $\mathcal{D}$ with the reduced arc set $\mathcal{A} \setminus \{(i, o^k) : (i, o^k) \in \mathcal{A}\} \setminus \{(d^k, j) : (d^k, j) \in \mathcal{A}\}$ computed with respect to travel times τ_{ij}. Given any k-feasible path P^k and two nodes i and j visited by the path, where j follows i in the path, value $\underline{\tau}^k(i, j)$ represents a valid lower bound on the difference between the arrival time at node j and the departure time at node i of path P^k. Further, for $i \in \mathcal{N}$, let $\vec{t_i}(t)$ be the next time point of point t in set $\mathcal{T}_i$ computed as

$$\vec{t_i}(t) = \begin{cases} \text{argmin}\{t' \in \mathcal{T}_i : t' > t\}, & \text{if } t < \text{argmax}\{t' \in \mathcal{T}_i\}, \\ \max_{k\in\mathcal{K}}(l^k - \underline{\tau}^k(i, d^k)), & \text{if } t = \text{argmax}\{t' \in \mathcal{T}_i\}, \end{cases} \tag{2.21}$$

where the term $l^k - \underline{\tau}^k(i, d^k)$ represents the latest departure time from node i for commodity k to arrive at its destination o^k before l^k. Function $\xi_i(t)$ for node $i \in \mathcal{N}$ and time point $t \in \mathcal{T}_i$, can be computed as:

$$\xi_i(t) = \begin{cases} \vec{t_i}(t), & \text{if } \vec{t_i}(t) - t > \hat{\Delta}, \\ t, & \text{if } \vec{t_i}(t) - t \leq \hat{\Delta}. \end{cases} \tag{2.22}$$

Function $\xi_i(t)$ computes an upper bound on time t and it returns t if the next time point in set $\mathcal{T}_i$ is at a distance in time less than or equal to the discretization level $\hat{\Delta}$, otherwise it returns either $\max_{k\in\mathcal{K}}(l^k - \underline{\tau}^k(i, d^k))$ or the next time point $t' \in \mathcal{T}_i$ such that $t' > t$. Figure 2.4 illustrates the computation of function $\xi_i(t)$.

Based on the above observations, we obtain the following relaxation of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$), called SND-HC-R($\mathcal{D}_{\mathcal{T}}$):

$$z_R(\mathcal{D}_{\mathcal{T}}) = \min \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} f_{ij}\cdot y_{ij}^{t\bar{t}} + \sum_{k\in\mathcal{K}} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} (c_{ij}^k q^k)\cdot x_{ij}^{kt\bar{t}} + \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}}(h_i^k q^k)\cdot w_i^k \tag{2.23}$$

$$s.t.\quad (2.5), (2.6), (2.8), (2.9), (2.10) \text{ and} \tag{2.24}$$

$$w_i^k \leq \begin{cases} \sum\limits_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} \xi_i(t) x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum\limits_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}} t\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum\limits_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} \xi_i(t) x_{ij}^{kt\bar{t}} - \sum\limits_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}} t\, x_{ji}^{k\bar{t}t}, & \text{otherwise}, \end{cases} \quad \forall\, i\in\mathcal{N}, \forall\, k\in\mathcal{K}, \tag{2.25}$$

$$\sum_{i\in\mathcal{N}} w_i^k = l^k - e^k - \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\hat{\mathcal{T}}}} \tau_{ij} x_{ij}^{kt\bar{t}}, \quad \forall\, k\in\mathcal{K}. \tag{2.26}$$

Inequalities (2.25) relax equations (2.7), whereas constraints (2.26) impose the total holding time for every commodity $k\in\mathcal{K}$. The following theorem holds.

Theorem 1. *If the partially time-expanded network $\mathcal{D}_{\mathcal{T}}$ satisfies Properties 1-3, then $z_R(\mathcal{D}_{\mathcal{T}}) \leq z(\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}})$, i.e., relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) is a valid SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$) relaxation.*

Proof. Let $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ be an optimal solution of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$) (i.e., an optimal CTSNDP-HC solution) of cost $\overline{z}$, and let $\overline{\mathcal{A}} = \{((i,t),(j,t+\tau_{ij})) \in \mathcal{A}_{\hat{\mathcal{T}}}^{\hat{\Delta}} : \overline{y}_{ij}^{t,t+\tau_{ij}} > 0\}$ be the set of arcs traversed by the commodities. Below we show that to solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ corresponds a feasible, but not necessarily optimal, solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) of cost $z = \overline{z}$.

By means of the mapping described by expressions (2.20), we can associate with vectors $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$ and corresponding paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$, solution vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. As shown by Boland et al. [20], to solution vector $\overline{\boldsymbol{x}}$ corresponds a set $\{P^k\}_{k\in\mathcal{K}}$ of feasible paths in network $\mathcal{D}_{\mathcal{T}}$, one path for each commodity, with the same total fixed and flow cost of paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$. More precisely, for each commodity $k\in\mathcal{K}$ and path

$\overline{P}^k = (a_1^k, \dots, a_{m^k}^k)$, $a_h^k \in \overline{\mathcal{A}}$, $h = 1, \dots, m^k$, with $a_h^k = ((i_h^k, t_h^k), (i_{h+1}^k, t_h^k + \tau_{i_h^k i_{h+1}^k}))$ and $t_{h+1}^k \geq t_h^k + \tau_{i_h^k i_{h+1}^k}$ for $h = 1, .., m^k - 1$ induced by solution vector $\overline{\boldsymbol{x}}$, corresponds a feasible path $P^k = (\mu(a_1^k), \dots, \mu(a_{m^k}^k))$ in $\mathcal{D}_{\mathcal{T}}$ with appropriate holding arcs. For each $k \in \mathcal{K}$ we have that the total transit time $T(P^k)$ of path P^k can be computed as

$$T(P^k) = T(\overline{P}^k) = \sum_{h=1}^{m^k} \tau_{a_h^k} = \sum_{((i,t),(j,\bar{t})) \in \overline{\mathcal{A}}} \tau_{ij} \overline{x}_{ij}^{kt\bar{t}} = \sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} \tau_{ij} x_{ij}^{kt\bar{t}}. \tag{2.27}$$

The holding times $\overline{\boldsymbol{w}}$ of solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ can be computed as:

$$\overline{w}_i^k = \begin{cases} t_1^k - e^k, & i = o^k, \\ l^k - (t_{m^k}^k + \tau_{i_{m^k}^k d^k}), & i = d^k, \\ t_h^k - (t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}), & i = i_h^k, h = 2, ..., m^k, \\ 0, & \text{otherwise}, \end{cases} , \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}.$$

It is easy to see that for each $k \in \mathcal{K}$ we have

$$\sum_{i \in \mathcal{N}} \overline{w}_i^k = l^k - e^k - T(\overline{P}^k). \tag{2.28}$$

We now show that solution $w_i^k = \overline{w}_i^k$, $\forall i \in \mathcal{N}$, $k \in \mathcal{K}$, satisfies constraints (2.25) and (2.26), thus showing that to solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ corresponds a feasible solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ of SND-HC-R($\mathcal{D}_{\mathcal{T}}$) of the same total fixed, flow and holding costs. According to (2.27) and (2.28) for all $k \in \mathcal{K}$, we have that constraints (2.26) are satisfied by solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$. Constraints (2.25) for the values $\boldsymbol{w}$ are as follows:

$$w_i^k \leq \begin{cases} \xi_{i_1^k}(\rho_{i_1^k}(t_1^k)) - e^k, & i = o^k, \\ l^k - \sigma(a_{m^k}^k), & i = d^k, \\ \xi_{i_h^k}(\rho_{i_h^k}(t_h^k)) - \sigma(a_{h-1}^k), & i = i_h^k, h = 2, ..., m^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}. \tag{2.29}$$

Based on the mapping functions $\rho(.)$, $\sigma(.)$ and $\xi(i)$ (see Figure 2.5), we have that for each $k \in \mathcal{K}$:

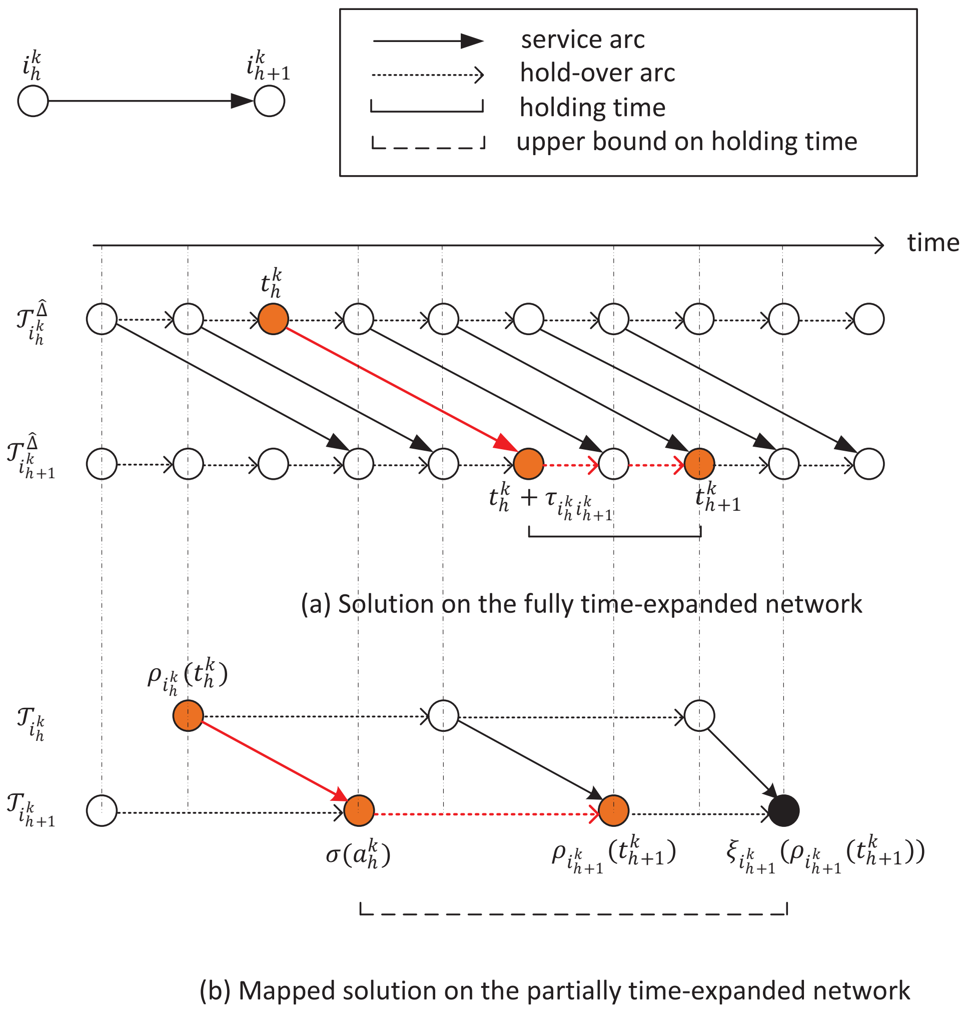


Figure 2.5 Illustration of the mapping functions $\rho(.)$, $\sigma(.)$ and $\xi(i)$

(i) $\xi_{i_h^k}(\rho_{i_h^k}(t_h^k)) \geq t_h^k$, for $h = 1, \dots, m^k$, and $\xi_{i_1^k}(\rho_{i_1^k}(t_1^k)) - e^k \geq t_1^k - e^k = \overline{w}_{o^k}^k$.

(ii) $\sigma(a_{h-1}^k) \leq t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}$ $h = 2, \dots, m^k$, and $l^k - \sigma(a_{m^k}^k) \geq l^k - (t_{m^k}^k + \tau_{i_{m^k}^k d^k}) = \overline{w}_{d^k}^k$.

(iii) $\xi_{i_h^k}(\rho_{i_h^k}(t_h^k)) - \sigma(a_{h-1}^k) \geq t_h^k - (t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}) = \overline{w}_{i_h^k}^k$, $h = 2, ..., m^k$.

(iv) For each $i \notin P^k$, we have $w_i^k \leq 0$, hence $w_i^k = 0$.

Solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ is then proved to be a feasible SND-HC-R($\mathcal{D}_{\mathcal{T}}$) solution with the same flow, fixed and holding costs, as well as the same total cost, as solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$. $\square$

The lower bound LB of the exact algorithm is thus computed as $LB = z_R(\mathcal{D}_\mathcal{T})$ where, in the computational results reported in Section 2.4, relaxation SND-HC-R$(\mathcal{D}_\mathcal{T})$ is solved by means of a general MIP solver.

2.3.3 Strengthening Relaxation SND-HC-R$(\mathcal{D}_\mathcal{T})$

The quality of relaxation SND-HC-R$(\mathcal{D}_\mathcal{T})$ strongly affects the effectiveness of the DDD algorithm for solving the CTSNDP-HC. In this section, we further describe ways to strengthen relaxation SND-HC-R$(\mathcal{D}_\mathcal{T})$ in order to obtain tighter lower bounds.

We first observe that since variables $\boldsymbol{w}$ are nonnegative, equations (2.26) also imply that

$$\sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} \tau_{ij} x_{ij}^{kt\bar{t}} \leq l^k - e^k, \quad \forall\, k \in \mathcal{K}, \tag{2.30}$$

that is, a path P^k for commodity $k \in \mathcal{K}$ from the origin o^k to the destination d^k cannot exceeds the maximum transit time computed as $l^k - e^k$.

As discussed by Boland et al. [20], the CTSNDP relaxation can be strengthened by means of the following additional *longest-feasibility-arc* property, based on which we can establish Theorem 2 below.

Property 4. *If arc $(i,t),(j,t') \in \mathcal{A}_\mathcal{T}$, then there does not exist a node $(j,t'') \in \mathcal{N}_\mathcal{T}$ with $t' < t'' \leq t + \tau_{ij}$, i.e., $t' = \arg max\{s : s \leq t + \tau_{ij}, (j,s) \in \mathcal{N}_\mathcal{T}\}$.*

Theorem 2. *For a fixed set of time points $\mathcal{T}$, (and thus fixed node set $\mathcal{N}_\mathcal{T}$), among the partially time-expanded networks $\mathcal{D}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T})$ satisfying Properties 1-3, consider the one $\overline{\mathcal{D}}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \overline{\mathcal{A}}_\mathcal{T})$ that also satisfies Property 4. We have:*

$$z_R(\overline{\mathcal{D}}_\mathcal{T}) \geq z_R(\mathcal{D}_\mathcal{T}).$$

Proof. Let $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ be a feasible solution of formulation SND-HC-R$(\overline{\mathcal{D}}_\mathcal{T})$ with the objective value $\overline{z}$. We show that to solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ corresponds a feasible, but not necessarily optimal, solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ of formulation SND-HC-R$(\mathcal{D}_\mathcal{T})$, with an objective value z such that $\overline{z} = z$.

Let $\overline{\mathcal{A}} = \{((i,t),(j,t')) \in \overline{\mathcal{A}}_{\mathcal{T}} : \overline{y}_{ij}^{tt'} > 0\}$ be the set of arcs traversed by solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$. Consider an arc $((i,t),(j,t')) \in \overline{\mathcal{A}}$ such that all arcs of the form $((i,t),(j,t''))$ belong to $\mathcal{A}_{\mathcal{T}}$ with $t'' < t'$. If no such arc $((i,t),(j,t'))$ exists, solution vectors $\boldsymbol{x} = \overline{\boldsymbol{x}}$ and $\boldsymbol{y} = \overline{\boldsymbol{y}}$ are clearly feasible for constraints (2.5), (2.6), (2.8), and (2.9) of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$). If such arc $((i,t),(j,t'))$ exists, since networks $\overline{\mathcal{D}}_{\mathcal{T}}$ and $\mathcal{D}_{\mathcal{T}}$ are defined based on the same set of time nodes $\mathcal{N}_{\mathcal{T}}$, there exists a path from (j,t'') to (j,t') in network $\mathcal{D}_{\mathcal{T}}$.

Initialize $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{y} = \mathbf{0}$ and define $x_{ij}^{kt\bar{t}} = \overline{x}_{ij}^{kt\bar{t}}$, $k \in \mathcal{K}$, and $y_{ij}^{t\bar{t}} = \overline{y}_{ij}^{t\bar{t}}$ for all arcs in $((i,t),(j,\bar{t})) \in (\mathcal{H}_{\mathcal{T}} \cup \mathcal{A}_{\mathcal{T}}) \cap (\overline{\mathcal{H}}_{\mathcal{T}} \cup \overline{\mathcal{A}}_{\mathcal{T}})$. We can adapt the solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$ with regard to the arc $((i,t),(j,t'))$ to the solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ concerning the arc $((i,t),(j,t''))$, with the addition of the holding arcs joining (j,t'') to (j,t')), by setting $y_{ij}^{tt''} = \overline{y}_{ij}^{tt'}$ and $x_{ij}^{ktt''} = x_{jj}^{kt''t'} = \overline{x}_{ij}^{ktt'}$. The resulting $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ solution is also feasible for constraints (2.5), (2.6), (2.8), and (2.9), and the process can be repeated for every arc $((i,t),(j,t')) \in \overline{A}_{\mathcal{T}}$ with $t'' < t'$ for all $((i,t),(j,t'')) \in A_{\mathcal{T}}$. For each commodity $k \in \mathcal{K}$, let $\overline{P}^k = (a_1^k, \ldots, a_{m^k}^k)$, $a_h^k \in \overline{\mathcal{A}}$, $h = 1, \ldots, m^k$, be the path induced by solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}})$ with $a_h^k = ((i_h^k, \overline{t}_h^k), (i_{h+1}^k, \overline{\pi}_{h+1}^k))$, $h = 1, \ldots, m^k$, where $\overline{t}_h^k$ is the departure time at node i_h^k and $\overline{\pi}_h^k$ is the corresponding arrival time, $h = 1, \ldots, m^k + 1$. Due to the definition of solution vectors $(\boldsymbol{x}, \boldsymbol{y})$ based on solution vectors $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}})$, in graph $\mathcal{D}_{\mathcal{T}}$ for commodity k we have a path $P^k = \overline{P}^k = (a_1^k, \ldots, a_{m^k}^k)$, $a_h^k \in \overline{\mathcal{A}}$, with departure time $t_h^k = \overline{t}_h^k$ and arrival times $\pi_h^k \leq \overline{\pi}_h^k$, $h = 1, \ldots, m^k + 1$.

We now show that solution vector $\boldsymbol{w} = \overline{\boldsymbol{w}}$ satisfies constraints (2.25) and (2.26) of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), thus showing that $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ is a feasible SND-HC-R($\mathcal{D}_{\mathcal{T}}$) solution having the same cost of solution $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{w}})$. First, for each $k \in \mathcal{K}$, $T(\overline{P}^k) = T(P^k)$, hence equations (2.26) are satisfied. Define $\overline{N}_k = \bigcup_{h=1,\ldots,m^k+1}\{i_h^k\}$, $k \in \mathcal{K}$, as the set of nodes visited by path $\overline{P}_k$. Then, we have that solution vector $\overline{\boldsymbol{w}}$ is defined as:

$$\overline{w}_i^k \leq \begin{cases} \xi_i(\bar{t}_1^k) - e^k, & i = o^k, \\ l^k - \overline{\pi}^k_{i^k_{m^k}}, & i = d^k, \\ \xi_i(\bar{t}_h^k) - \overline{\pi}_h^k, & \text{if } i = i_h^k \in \overline{N}_k \setminus \{o^k, d^k\}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall\, i \in \mathcal{N}.$$

and for inequalities (2.25) we have

$$w_i^k \leq \begin{cases} \xi_i(t_1^k) - e^k = \xi_i(\bar{t}_1^k) - e^k, & i = o^k, \\ l^k - \pi^k_{i^k_{m^k}} \geq l^k - \overline{\pi}^k_{i^k_{m^k}}, & i = d^k, \\ \xi_i(t_h^k) - \pi_h^k \geq \xi_i(\bar{t}_h^k) - \overline{\pi}_h^k, & \text{if } i = i_h^k \in \overline{N}_k \setminus \{o^k, d^k\}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall\, i \in \mathcal{N}.$$

Hence, $\boldsymbol{w} = \overline{\boldsymbol{w}}$ is proved to be a feasible solution for inequalities (2.25). $\square$

2.3.4 Initial Partially Time-Expanded Network

Without loss of generality, we assume that $min_{k\in\mathcal{K}}\{e^k\} = 0$. The initial partially time-expanded network $\mathcal{D}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T})$ is defined in order to satisfy Properties 1-4 as follows:

- According to Property 1, for all $k \in \mathcal{K}$, nodes (o^k, e^k) and (d^k, l^k) are included in node set $\mathcal{N}_\mathcal{T}$.
- According to Properties 2 and 3, for each $i \in \mathcal{N}$, a node $(i, 0)$ is added to $\mathcal{N}_\mathcal{T}$. Moreover, based also on Property 4, for each node $(i,t) \in \mathcal{N}_\mathcal{T}$ and for each arc $(i,j) \in \mathcal{A}$, arc $((i,t),(j,t'))$ with $t' = \text{argmax}\{s \in \mathcal{T}_j : s \leq t + \tau_{ij}\}$ is added to arc set $\mathcal{A}_\mathcal{T}$.
- For each $(i,t) \in \mathcal{N}_\mathcal{T}$, a holding arc $((i,t),(i,t'))$ is added to set $\mathcal{H}_\mathcal{T}$ if $t' = \text{argmin}\{s \in \mathcal{T}_i : s > t\}$ exists.

Function $\xi_i(t)$, $\forall i \in \mathcal{N}$, $t \in \mathcal{N}_{\mathcal{T}}$, is initialed based on the initial partially time-expanded network and expressions (2.22).

In order to keep the number of variables and constraints of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) as small as possible, we use the reduction rules described in Marshall et al. [81]. These rules are based on shortest path calculations and can identify the variables and constraints that cannot be part of any optimal SND-HC-R($\mathcal{D}_{\mathcal{T}}$) solution. The reader may refer to Marshall et al. [81] for additional details.

2.3.5 Computing a Feasible CTSNDP-HC solution

Our exact algorithm computes an upper bound UB on the optimal solution cost of CTSNDP-HC based on the implementable model $IM(\mathcal{S})$ described in Section 2.2.3. More precisely, the flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$ computed by solving relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) is used to derive a feasible CTSNDP-HC solution. Since SND-HC-R($\mathcal{D}_{\mathcal{T}}$) incorporates holding costs, our upper bound heuristic method also incorporates holding costs.

For the CTSNDP, where holding costs are all zero, a solution $(\boldsymbol{\pi}, \boldsymbol{t}, \hat{\boldsymbol{t}})$ satisfying (2.12)-(2.19) together with the flat solution $\mathcal{S}$ corresponds to a feasible CTSNDP solution of cost $z_{fc}(\mathcal{S}) = LB$. Thus, as observed by Boland et al. [20], the optimality for the CTSNDP can be proved whenever the flat solution $\mathcal{S}$ associated with the relaxation is implementable. Hence, problem $IM(\mathcal{S})$ for the CTSNDP reduces to a feasibility problem. Marshall et al. [81] investigated the structure of the flat solution $\mathcal{S}$ for the CTSNDP as a graph theoretical problem, and identified two cases inducing non-implementable flat solutions (see Lemmas 1 and 2 of Marshall et al. [81]).

For the CTSNDP-HC, due to the presence of the in-storage holding costs and the approximation introduced by relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), an implementable solution $\mathcal{S}$ can correspond to a feasible solution with cost $UB = z_{fc}(\mathcal{S}) + z_w(\mathcal{S})$ greater than the cost of the current lower bound LB, and optimality cannot be proved. However, clearly, if $UB = LB$, then an optimal CTSNDP-HC solution has been identified.

The procedure performs the following two steps: (i) formulation $IM(\mathcal{S})$ is solved without constraints (2.16) in order to identify infeasible consolidation constraints (2.14) which are related to Lemma 1 of Marshall et al. [81] and are selectively removed from the formulation; and (ii) the resulting updated $IM(\mathcal{S})$ model with constraints (2.16) and a reduced set of constraints (2.14) is solved in order to identify additional infeasible consolidation constraints which are related to Lemma 2 of Marshall et al. [81].

The procedure identifies infeasible consolidation constraints by computing an irreducibly inconsistent system (IIS) of formulation $IM(\mathcal{S})$, which is a description of the minimal subproblem that is still infeasible [101, 32]. An infeasible subproblem is minimal if, when any of the constraints are removed, the infeasibility vanishes, and algorithms for identifying IIS have been investigated in Gleeson and Ryan [57] and Chinneck [31]. Algorithm 2 gives the steps of the procedure. In the algorithm, sets $\mathcal{J}$, $\mathcal{J}_C$ and $\mathcal{J}_P$ represent index sets associated with consolidation constraints (2.14). Function `LP-Solve(.)` iteratively solves an LP model until a feasible solution is found. The function updates the set of consolidation constraints $\mathcal{J}$ and determines the set of infeasible consolidations $\overline{\mathcal{J}}$. Algorithm 2 first identifies the set $\mathcal{J}_C$ of infeasible consolidation constraints (line 13) and then the additional set of infeasible constraints $\mathcal{J}_P$ (line 16). The final UB value is computed based on the final subset of consolidation constraints identified, and $\mathcal{J}_C$ and $\mathcal{J}_P$ help identify how to refine the current partially time-expanded network.

2.3.6 Refining a Partially Time-Expanded Network

If our exact algorithm (adapted from Algorithm 1 in Section 2.3.1) does not terminate, then the upper bound UB computed by Algorithm 2 is greater than the current lower bound LB, and the corresponding gap is greater than the given optimality tolerance. In this case, at least one of the following two cases applies:

(i) The flat solution $\mathcal{S}$ is proved to be non-implementable, and at least one of the sets $\mathcal{J}_C$ and $\mathcal{J}_P$ is non-empty. This implies that in the solution obtained by the relaxation model, there is at least one commodity $k \in \mathcal{K}$ routed on an arc

Algorithm 2: Upper bound computation and identification of the sets of infeasible consolidation constraints

Input: Flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$
Output: Upper bound UB and consolidation constraint subsets $\mathcal{J}_C$ and $\mathcal{J}_P$
Function `Solve-LP(`LP, $\mathcal{J}$, $\overline{\mathcal{J}}$`)`:
  begin
    while *not solved* **do**
      Solve problem LP with consolidation constraints in $\mathcal{J}$;
      if *LP is infeasible* **then**
        Detect an IIS and select from the IIS $\mathcal{J}' \subseteq \mathcal{J}$ and compute $r^* = \text{argmin}\{f_{\alpha_r} : r \in \mathcal{J}'\}$;
        $\mathcal{J} \leftarrow \mathcal{J} \setminus \{r^*\}$ and $\overline{\mathcal{J}} \leftarrow \overline{\mathcal{J}} \cup \{r^*\}$;
      end
    end
    return $\mathcal{J}$ *and* $\overline{\mathcal{J}}$;
  end
begin
  `// Initialization`
  $\mathcal{J} \leftarrow \{1, \ldots, |\mathcal{C}|\}$, $\mathcal{J}_C \leftarrow \emptyset$ and $\mathcal{J}_P \leftarrow \emptyset$;
  `// Compute set` $\mathcal{J}_C$ `associated with cycles`
  Let LP be model $IM(\mathcal{S})$ without constraints (2.14) and (2.16);
  `Solve-LP(`LP, $\mathcal{J}$, $\mathcal{J}_C$`)`;
  `// Update the set of consolidation constraints`
  $\mathcal{C} \leftarrow \{\mathcal{C}_r : r \in \mathcal{J}\}$;
  `// Compute set` $\mathcal{J}_P$ `associated with paths`
  Let LP be the model $IM(\mathcal{S})$ without constraints (2.14);
  `Solve-LP(`LP, $\mathcal{J}$, $\mathcal{J}_P$`)`;
  `// Update the set of consolidation constraints`
  $\mathcal{C} \leftarrow \{\mathcal{C}_r : r \in \mathcal{J}\}$;
  `// A feasible solution has been identified`
  $UB \leftarrow z_{fc}(\mathcal{S}) + z_w(\mathcal{S})$;
  return UB, $\mathcal{J}_C$ *and* $\mathcal{J}_P$;
end

$((i,t),(j,t'))$ that is too short, i.e., $t' < t + \tau_{ij}$. We call an arc $((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$ such that $t' < t + \tau_{ij}$ a *short-arc*. The reason for the non-implementability is due to the fact that short-arcs are evaluated by model $IM(\mathcal{S})$ with the actual or true travel times τ_{ij}. In this case, the short-arcs identified must be lengthen by adding new time points to network $\mathcal{D}_\mathcal{T}$. Network $\mathcal{D}_\mathcal{T}$ is also updated in such a way that the current flat solution $\mathcal{S}$ is no more feasible for the relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) defined on the updated network $\mathcal{D}_\mathcal{T}$.

(ii) The relaxation of the holding variables $\boldsymbol{w}$ defined by inequalities (2.25) is too weak, i.e., the upper bounds on the departure times computed by functions $\xi_i(.)$

and the lower bounds on the arrival times computed by functions $\sigma(.)$ must be strengthened.

We observe that if the flat solution $\mathcal{S}$ is implementable, a sufficient condition for the solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ of the relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) to corresponds to an optimal CTSNDP-HC solution of cost LB is that $w_i^k = \theta_i^k$, $\forall k \in \mathcal{K}$, $i \in \mathcal{N}$, where:

$$\theta_i^k = \begin{cases} \sum\limits_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} t\, x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum\limits_{((j,\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} t\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum\limits_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} t\, x_{ij}^{kt\bar{t}} - \sum\limits_{((j,\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} t\, x_{ji}^{k\bar{t}t}, & \text{otherwise}, \end{cases} \qquad \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}. \tag{2.31}$$

If $\boldsymbol{w} \neq \boldsymbol{\theta}$, it exists at least one commodity k passing through a node i such that $w_i^k > \theta_i^k$, and at least one of the following two cases applies:

(a) The upper bounds on the departure times for node i computed by function $\xi_i(.)$ must be reduced. Let $((i,t),(j,t'))$ be an arc traversed by the path associated with commodity k, and let $b = \xi_i(t)$ be the time point computed with respect to the current $\mathcal{D}_\mathcal{T}$ network. To strengthen the relaxation, a new time $\bar{t}$ must be added to network $\mathcal{D}_\mathcal{T}$ such that the function $\xi_i(t)$ is updated in $\xi_i(t) = \bar{t} < b$. A time point $\bar{t} = (i, t + \max\{\lfloor(w_i^k - \theta_i^k)/2\rfloor, 1\})$ suffices to the case.

(b) The lower bounds on the arrival times defined by the term $\sum_{((j',\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} t\, x_{j',i}^{k\bar{t}t}$ of equations (2.31) for the cases $i \neq o^k$ must be increased. This can be accomplished by lengthening additional short-arcs (if any).

Clearly, for the CTSNDP-HC with zero in-storage holding costs, this case (ii) does not apply, and in the worst case the set of nodes $\mathcal{N}_\mathcal{T}$ corresponds to set $\mathcal{N}_\mathcal{T}^\Delta$ with $\Delta = 1$.

Algorithm 3: Refinement of network $\mathcal{D}_\mathcal{T}$ with a new time point

Input: Network $\mathcal{D}_\mathcal{T}$ and time point t for node i
Output: Updated network $\mathcal{D}_\mathcal{T}$
1 **Function** `Refine`($\mathcal{D}_\mathcal{T}$, i, t):
  begin
2     $\mathcal{N}_\mathcal{T} \leftarrow \mathcal{N}_\mathcal{T} \cup \{(i,t)\}$;
    `// Add timed-copy arcs to node` (i,t) `satisfying Property 4`
3   **forall** $(i,j) \in \mathcal{A}$ **do**
4       $t' \leftarrow \text{argmax}\{s \in \mathcal{T}_j : s \leq t + \tau_{ij}\}$;
5       $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \cup ((i,t),(j,t'))$;
6   **end**
    `// Change the arcs to impose Property 4`
7   **forall** $((j,\bar{t}),(i,t')) \in \mathcal{A}$ *where* $t' = \text{argmax}\{s \in \mathcal{T}_i : s < t\}$ **do**
8   **if** $\bar{t} + \tau_{ij} \geq t$ **then**
9         $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \setminus \{((j,\bar{t}),(i,t'))\}$;
10         $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \cup \{((j,\bar{t}),(i,t))\}$;
11   **end**
12   **end**
    `// Update holding arc set`
13     $t' \leftarrow \text{argmax}\{s \in \mathcal{T}_i : s < t\}$ and $t'' \leftarrow \text{argmin}\{s \in \mathcal{T}_i : s > t\}$;
14     $\mathcal{H}_\mathcal{T} \leftarrow \mathcal{H}_\mathcal{T} \setminus \{((i,t'),(i,t''))\}$;
15     $\mathcal{H}_\mathcal{T} \leftarrow \mathcal{H}_\mathcal{T} \cup \{((i,t'),(i,t)),((i,t),(i,t''))\}$;
16   **return** $\mathcal{D}_\mathcal{T}$;
17   **end**

In the following, we describe how the network $\mathcal{D}_\mathcal{T}$ can be updated with new times points based on the aforementioned cases (i) and (ii).

Adding new time points to the partially time-expanded network $\mathcal{D}_\mathcal{T}$

The algorithm adopted to update the partially time-expanded network $\mathcal{D}_\mathcal{T}$ with a new time point t for node i follows the steps of a similar algorithm used by the DDD algorithm for the CTSNDP by Boland et al. [20]. Algorithm 3 gives the steps performed to update a network $\mathcal{D}_\mathcal{T}$ with a new time point t for node i. In the algorithm, time node (i,t) is first added to the set of time nodes $\mathcal{N}_\mathcal{T}$ (line 2). As required by Properties 3 and 4, for each arc $(i,j) \in \mathcal{A}$, the longest timed-copy arc associated with time node (i,t) is then added to the arc set $\mathcal{A}_\mathcal{T}$ (line 5). Whenever an arc with which Property 4 is not satisfied because of the addition of node (i,t), the network is updated by removing the arcs not satisfying Property 4 and by adding new arcs that do satisfy Property 4 (line 10). Finally, the set of holding arcs is updated (lines 13-15).

Algorithm 4: Refine strategy

Input: Network $\mathcal{D}_\mathcal{T}$ and consolidation constraint sets $\mathcal{J}_C$ and $\mathcal{J}_P$
Parameters: NA number of additional short-arcs to be lengthen
Output: Updated network $\mathcal{D}_\mathcal{T}$
begin
  `// Strategy 1: Lengthen short-arcs based on cycles`
  forall $r \in \mathcal{J}_C$ **do**
    forall $(k, n) \in J_r$ **do**
      $P^k \leftarrow$ path associated with commodity k;
      forall $((i,t),(j,t')) \in P^k$ **do**
        if $t' < t + \tau_{ij}$ **then** `Refine(`$\mathcal{D}_\mathcal{T}$,j,$t+\tau_{ij}$`)`;
      end
    end
  end
  `// Strategy 2: Lengthen short-arcs based on paths`
  forall $r \in \mathcal{J}_P$ **do**
    forall $(k, n) \in J_r$ **do**
      $P^k \leftarrow$ path associated with commodity k;
      $((i,t),(j,t')) \leftarrow a_n^k$;
      if $t' < t + \tau_{ij}$ **then** `Refine(`$\mathcal{D}_\mathcal{T}$,j,$t+\tau_{ij}$`)` ;
      $((i,t),(j,t')) \leftarrow \text{argmax}_{((i_1,t_1),(i_2,t_2))\in P^k}\{t_1 + \tau_{i_1,i_2} - t_2\}$;
      if $t' < t + \tau_{ij}$ **then** `Refine(`$\mathcal{D}_\mathcal{T}$,j,$t+\tau_{ij}$`)`;
    end
  end
  `// Strategy 3: Lengthen short-arcs to strengthen the relaxation`
  Sort the arcs $a = ((i,t),(j,t'))$ associated with $\{P^k\}_{k\in\mathcal{K}}$ having $t' < t + \tau_{ij}$ for increasing values of $\gamma_a = t$;
  $\mathcal{L} \leftarrow (a_1, a_2, \ldots, a_{|\mathcal{A}_\mathcal{T}|})$ such that $\gamma_{a_1} \leq \gamma_{a_2} \leq \cdots \leq \gamma_{a_{|\mathcal{A}_\mathcal{T}|}}$;
  forall $h \leftarrow 1$ *to* NA **do**
    $((i,t),(j,t')) \leftarrow a_h$;
    `Refine(`$\mathcal{D}_\mathcal{T}$,j,$t+\tau_{ij}$`)`
  end
  `// Strategy 4: Add new time points to strengthen the relaxation`
  forall $k \in \mathcal{K}$ **do**
    $P^k \leftarrow$ path associated with commodity k;
    forall $((i,t),(j,t')) \in P^k$ **do**
      if $w_i^k > \theta_i^k$ **then**
        `Refine(`$\mathcal{D}_\mathcal{T}$,j,$t + \max\{\lfloor (w_i^k - \theta_i^k)/2 \rfloor, 1\}$`)`;
      end
    end
  end
end

Notice that the value of functions $\xi_i(.)$ change according to the updated network $\mathcal{D}_{\mathcal{T}}$ and expressions (2.22).

Refine strategy

The refinement strategy identifies new time points to update the network $\mathcal{D}_{\mathcal{T}}$ in order to strengthen the relaxation. It is based on sets $\mathcal{J}_C$ and $\mathcal{J}_P$ of infeasible consolidation constraints obtained from the computation of a valid upper bound by Algorithm 2.

An overview of the steps executed by the refinement strategy is given by Algorithm 4. In the algorithm, short-arcs associated with the set of infeasible consolidations $\mathcal{J}_C$ are first identified (*Strategy 1*, lines 1-8), followed by lengthening of the short-arcs associated with infeasible consolidations from set $\mathcal{J}_P$ (*Strategy 2*, lines 9-17). To strengthen the relaxation of the holding variables $\boldsymbol{w}$, the algorithm lengthens the short-arcs associated with paths $\{P^k\}_{k\in\mathcal{K}}$ selected in the flat solution (*Strategy 3*, lines 18-23). It is worth noting that the paths composing an implementable flat solution can contain short-arcs, hence in this case Strategies 1 and 2 are not used since $\mathcal{J}_C = \mathcal{J}_P = \emptyset$, and the short-arcs are lengthened by Strategy 3. Finally, new time points are added by *Strategy 4* (lines 24-31).

It is worth noting that with zero in-storage holding costs, refinement Strategies 1 and 2 suffice for the convergence of the algorithm, since case (ii) of Section 2.3.6 does not apply.

2.3.7 Convergence and Optimality

Given the domains of the $\boldsymbol{x}$ and $\boldsymbol{y}$ variables of formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$) defined on the fully time-expanded network $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$, there are only a finite number of solutions corresponding to all feasible solution vectors $(\boldsymbol{x}, \boldsymbol{y})$ of the formulation.

At the different iterations of the algorithm, according to our proposed refinement process, if a flat-solution is non-implementable, at least one short-arc is identified by Algorithm 2 and is lengthened by the refinement process. Moreover, if the current relaxation of the holding variables $\boldsymbol{w}$ is too weak, new additional short-arcs are lengthened and new time points are added to strengthen the relaxation. As a result of the refinement process, the current solution of the relaxation is no longer feasible for the updated relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$).

Under our assumption made in Section 2.2.1, the CTSNDP-HC always has a feasible solution. Therefore, since there are only finitely many feasible solutions, arcs to be lengthened and time points to be added, repeating this process finitely many times will guarantee the convergence of the solution associated with lower bound LB to an optimal solution of the CTSNDP-HC. We have thus established the following result.

Theorem 3. *If the optimality tolerance is set equal to zero, the DDD algorithm for the CTSNDP-HC converges to an optimal solution after finitely many iterations.*

2.4 Computational Experiments

In this section, we present extensive computational analysis based on two sets of experiments. Through the first set of experiments (see Section 2.4.1), we evaluate the performance of the DDD algorithm for the CTSNDP-HC on instances derived from the literature. Through the second set of experiments (see Section 2.4.2), we examine the quality of the solutions produced on an additional set of instances, so as to analyze the effectiveness of the DDD algorithm and identify factors that affect the complexity of the CTSNDP-HC. Based on the results from these two sets of experiments, we evaluate the importance and benefits of taking holding costs into account in solving the CTSNDP.

We denote with EXM our DDD algorithm for the CTSNDP-HC. In order to further compare the performance of EXM, we also implemented the algorithm proposed by Boland et al. [20] for the CTSNDP, hereafter denoted as EXM-0. Algorithm EXM-0 was used as a baseline algorithm to compute heuristic solutions for the CTSNDP-HC so as to analyze the performance of algorithm EXM. It is worth noting that DDD-based methods rely on two (relative) optimality tolerances: (i) the optimality tolerance used by the DDD algorithm (see parameter *optimality tolerance* of the exact algorithm described in Section 2.3.1), and (ii) the optimality tolerance used by the MIP solver. For the sake of the notation, we denote by tol_{DDD} and tol_{MIP} the two optimality tolerances, respectively. Given tolerance tol_{MIP} applied to the MIP solver and in order to compute safe lower bounds, the lower bound value LB was set equal to the best known lower bound on the optimal objective given by the Gurobi solver at termination through parameter `ObjBoundC`.

The algorithms were implemented in Java language, and Gurobi (v.8.1.1) [60] was used as the LP solver to solve model $IM(\mathcal{S})$, and as the MIP solver to solve relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$). The Gurobi function `Model.computeIIS()` was used to compute IISs in Algorithm 2. The experiments were performed on an Intel(R) Core(TM) i7-8700 (3.20 GHz) Desktop PC equipped with 64 GB RAM running under Windows 10 64-bit operating system.

2.4.1 Experiments based on CTSNDP Benchmark Instances

In our first set of experiments, we aim to test the performance of EXM in solving CTSNDP benchmark instances used in the literature and CTSNDP-HC instances generated from these CTSNDP benchmark instances.

Instance Generation

We considered the set of 558 CTSNDP instances generated by Boland et al. [21] that also contains the 432 instances used by Boland et al. [20]. The 558 instances were also used by Marshall et al. [81] to obtain their computational results.

The instances were derived from the 31 classes of the "C" instances presented in Crainic et al. [34], which have been widely used in the literature as benchmarks to evaluate the solution methods for the capacitated fixed charge network design problem [56, 64, 36, 65]. Table 2.2 gives the details of the classes of networks $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ considered by Crainic et al. [34]. In the table, the column "Cost ratio" computed as $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \frac{f_a}{c_a u_a}$ measures the ratio between fixed and variable costs, "Cap ratio" computed as $\sum_{k\in\mathcal{K}} q^k / \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} u_a$ indicates whether the arcs are loosely or tightly capacitated,

Table 2.2 Characteristics of "untimed" **C** instances

Class	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{K}\|$	Cost ratio	Cap ratio	Avg length
c33	20	228	39	0.02	5.8	2407.9
c35	20	230	40	0.02	16.0	767.9
c36	20	230	40	0.08	16.0	3705.8
c37	20	228	200	0.51	16.0	1871.4
c38	20	230	200	0.97	16.0	4381.0
c39	20	229	200	0.47	20.0	1691.3
c40	20	228	200	0.94	22.0	3522.1
c41	20	288	40	0.02	8.0	1622.0
c42	20	294	40	0.08	10.0	5675.8
c43	20	294	40	0.02	16.0	776.5
c44	20	294	40	0.08	16.0	3517.9
c45	20	294	200	0.48	25.0	1124.2
c46	20	292	200	1.01	25.0	2632.0
c47	20	291	200	0.46	28.0	996.6
c48	20	291	200	0.95	28.0	2271.6
c49	30	518	100	0.10	20.0	341.1
c50	30	516	100	0.51	20.0	1586.5
c51	30	519	100	0.09	29.9	206.6
c52	30	517	100	0.49	29.9	1161.5
c53	30	520	400	0.18	40.0	612.1
c54	30	520	400	0.36	40.0	1061.8
c55	30	516	400	0.18	49.9	479.4
c56	30	518	400	0.35	49.9	966.9
c57	30	680	100	0.09	20.0	307.6
c58	30	680	100	0.20	20.0	592.8
c59	30	687	100	0.10	29.9	187.1
c60	30	686	100	0.20	29.9	394.7
c61	30	685	400	0.19	40.0	503.8
c62	30	679	400	0.36	40.0	1056.5
c63	30	678	400	0.18	49.9	381.4
c64	30	683	400	0.34	49.9	780.0

and "Avg length" computed as $\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \mathcal{I}_{o^k d^k}$, where $\mathcal{I}_{o^k d^k}$ is the length of the least total travel time path from o^k to d^k, is the average length of the least total travel time paths. For each of the 31 classes of networks reported in the table, Boland et al. [21] generated 18 CTSNDP timed instances by first calculating the travel times for each arc and then by generating the time windows for each commodity by randomly sampling from a normal distribution. Based on Boland et al. [22] and as also reported by Marshall et al. [81], these instances can also be grouped by the flexibility and cost ratio of the instances, these being a measure of the tractability of the instances. An instance has (i) *low flexibility* (LF) if $\min_{k\in\mathcal{K}}\{l^k - (e^k + \mathcal{I}_{o^k d^k})\} < 227$, and *high* *flexibility* (HF) otherwise, and (ii) *low cost* ratio (LC) if $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\frac{f_a}{c_a u_a} < 0.175$, and *high cost* ratio (HC) otherwise. The instances are then grouped according to the two measures, resulting in the four groups of instances, referred to as "HC/LF", "HC/HF", "LC/LF" and "LC/HF", respectively.

For each of the 558 CTSNDP instances, we generated a CTSNDP-HC instance by first computing for each node $i \in \mathcal{N}$ parameter $\epsilon_i = 1/|\mathcal{A}_i|\sum_{a\in A_i}(c_a + f_a/u_a)/\tau_a$, where $\mathcal{A}_i = \{(j_1, i), (i, j_2) : (j_1, i), (i, j_2) \in \mathcal{A}\}$ represents the average fixed and variable costs of all the ingoing and outgoing arcs of node i. Then, the per-unit-of-demand-and-time holding cost h_i^k was set equal to $0.3\,\epsilon_i$, $\forall k \in \mathcal{K}$. Value 0.3 was chosen to simulate the fact that holding costs typically range between 20% and 30% of the inventory value [85, 88], and the transportation cost is generally less than the commodity value. Since in practice a commodity does not incur any holding costs at the destination, for each commodity $k \in \mathcal{K}$ we set $h_{d^k}^k = 0$. We therefore generated 558 CTSNDP-HC instances.

Results

Before evaluating the performances of EXM , to confirm the effectiveness of the baseline algorithm, here we first report on a comparison of the results obtained by the baseline algorithm EXM-0 in solving the 558 CTSNDP instances with the results of the DDD algorithms proposed by Boland et al. [20], denoted as BHMS17, and Marshall et al. [81], denoted as MBSH21. A time limit of one hour was imposed to EXM-0, as done for both BHMS17 and MBSH21. In the comparison reported by Marshall et al. [81], for method BHMS17, tol_{DDD} and tol_{MIP} were set equal to 0.01 and 0.0001 (i.e., the CPLEX default setting), respectively. For method MBSH21, tol_{DDD} was also set equal to 0.01, whereas tolerance tol_{MIP} was dynamically changed during the different iterations. More precisely, the initial tolerance was set equal to 0.04, and then for each iteration tol_{MIP} was computed as $\max\{gap \times 0.25, tol_{DDD} \times 0.98\}$, where *gap* is the final gap of the previous iteration. Regarding EXM-0, tol_{DDD} and tol_{MIP} were set equal to 0.01 and 0.01, respectively. For the set of CTSNDP instances, our comparison is based on the results reported by Marshall et al. [81] which were obtained on a single core machine using CPLEX 12.6 as the MIP solver (for both BHMS17 and MBSH21, with no specific machine type being reported). Because the computational environment of BHMS17 and MBSH21 was different from that of our algorithms, a direct comparison is therefore not possible. However, in what follows we give a clear overall picture of the relative performance, especially when the total number of instances solved to proven optimality is compared.

Table 2.3 gives the comparison of the three algorithms. For each group of instances, the table shows the number of instances in the group and, for each algorithm, the average percentage deviation of the final upper bound UB computed with respect to the final lower bound LB ("$\%UB$"), i.e., $100.0 \times \frac{UB-LB}{UB}$, the average computing time in seconds ("*time*"), the average number of iterations ("*iter*"), and the percentage of the instances solved to optimality ("$\%opt$") (within the given optimality tolerance). The average values shown in columns "$\%UB$", "*time*"and "*iter*" were computed over all instances.

The table shows that our implementation of the algorithm of Boland et al. [20] compares well with both algorithms BHMS17 and MBSH21, and also shows similar performances on the different groups of instances. We note that the computational environment of EXM-0 is different from the one used by the other methods. However, the comparison over the total number of instances solved to proven optimality gives a clear global picture of the relative performance. In groups **LC/LF** and **LC/HF**, EXM-0 was capable of solving to optimality all the instances within the imposed optimality tolerance. In these groups, with respect to BHMS17 and MBSH21, EXM-0 shows higher percentage deviations of the final upper bound UB. This can be due to the fact that different MIP solvers are used, and that EXM-0 computes a safe lower bound based on the best known bound given by the Gurobi MIP solver. Based on the results of Table 2.3, we adopted algorithm EXM-0 as a baseline algorithm for comparison purposes with EXM.

Table 2.3 Summary results on the CTSNDP instances

Group	Algorithm	$\%UB$	*time*	*iter*	*%opt*
HC/LF	BHMS17	0.08	1391.1	5.3	77.1
183	MBSH21	0.12	677.8	14.8	85.8
	EXM-0	0.78	318.5	12.1	95.6
HC/HF	BHMS17	0.56	1966.7	6.0	53.7
177	MBSH21	0.84	1693.8	17.5	56.5
	EXM-0	3.31	1613.2	11.6	60.5
LC/LF	BHMS17	0.00	28.6	3.7	100.0
94	MBSH21	0.00	0.6	6.5	100.0
	EXM-0	0.62	0.8	3.7	100.0
LC/HF	BHMS17	0.00	1.5	2.5	100.0
104	MBSH21	0.00	0.1	3.2	100.0
	EXM-0	0.50	0.1	1.5	100.0

Table 2.4 Summary results on the CTSNDP-HC instances

	Zero holding costs														Nonzero holding costs										
	EXM-0							EXM							EXM										
		$\%UB0$							$\%UB$							$\%UB$						$\%LB0$	$\%UB1$	
Group	*%opt*	*min*	*max*	*avg*	*time*	$\%tLB$	*iter*	*%opt*	*min*	*max*	*avg*	*time*	$\%tLB$	*iter*	*%opt*	*min*	*max*	*avg*	*time*	$\%tLB$	*iter*	*avg*	*avg*	*max*
HC/LF	96.7	2.4	5.2	3.7	450.6	88.6	12.2	93.4	1.0	2.8	1.7	1170.9	93.3	7.5	78.1	1.0	4.5	1.7	2544.0	94.1	15.9	1.1	0.7	2.5
HC/HF	70.6	1.4	23.6	7.8	2878.0	95.1	12.7	68.9	1.2	8.8	2.9	3125.5	97.7	7.6	55.9	1.2	10.3	3.6	3786.3	97.7	10.0	1.7	1.5	14.5
LC/LF	100.0	-	-	-	0.8	60.0	3.7	100.0	-	-	-	1.3	68.0	3.1	100.0	-	-	-	1.9	70.7	4.7	0.5	1.3	2.8
LC/HF	100.0	-	-	-	0.1	40.9	1.5	100.0	-	-	-	0.1	51.7	1.6	100.0	-	-	-	0.2	55.2	2.4	0.2	1.5	3.1

Note: "-" represents that all instances in the corresponding instance group are solved to optimality by the corresponding method.

For the set of 558 CTSNDP-HC instances, we executed EXM with a time limit of two hours, and we also used EXM-0 to solve the corresponding CTSNDP instances and to compute heuristic solutions for the CTSNDP-HC, again with a time limit of two hours. In order to attest to the effectiveness of EXM in solving the CTSNDP instances, we also used algorithm EXM to solve the set of CTSNDP instances, that is, EXM was used by setting the holding costs equal to zero. For these experiments, we used $tol_{DDD} = 0.01$ for both EXM and EXM-0, $tol_{MIP} = 0.01$ for EXM-0. For algorithm EXM, based on our preliminary experiments, we also found to be computationally convenient to dynamically change parameter tol_{MIP}, that was computed as $\max\{\min\{0.04, gap \times 0.25\}, 0.01\}$.

Table 2.4 reports the corresponding results based on the categories of the flexibility and cost ratio of the instances. The following notation is used:

- $LB0$, $UB0$: final lower and upper bounds computed by EXM-0, respectively.
- $UB1$: value of the CTSNDP-HC solution derived from the upper bound $UB0$ computed by algorithm EXM-0, obtained by adding the holding costs associated with the holding times of the solution corresponding to $UB0$.
- LB, UB: final lower and upper bounds computed by EXM, respectively.

For each method and group of instances, the table shows the percentage of instances solved to optimality ("%*opt*"), the average percentage deviation of lower bound LB with respect to lower bound $LB0$ (i.e., $\%LB0 = 100.0 \times \frac{LB-LB0}{LB0}$) and the percentage deviations of the different upper bounds computed as $\%UB0 = 100.0 \times \frac{UB0-LB0}{UB0}$, $\%UB1 = 100.0 \times \frac{UB1-UB}{UB1}$ and $\%UB = 100.0 \times \frac{UB-LB}{UB}$. For the different percentage deviations, the *min* and *max* values are also reported. For $UB0$ and UB, the deviations were computed over all instances not solved to optimality, whereas the deviations of $UB1$ were computed over all instances. Columns *time* and *iter* give the average computing time and number of iterations, respectively, computed over all instances. Column $\%tLB$ reports the percentage of the total time spent in computing the lower bound, i.e., the percentage of the total time spent by the MIP solver over the total computing time.

Table 2.5 Summary results on the CTSNDP-HC instances by network class

		Zero holding costs								Nonzero holding costs						
		EXM-0				EXM				EXM						
			%$UB0$				%UB				%UB			%$LB0$	%$UB1$	
Class	*ni*	*opt*	*min*	*max*	*avg*	*opt*	*min*	*max*	*avg*	*opt*	*min*	*max*	*avg*	*avg*	*avg*	*max*
c33	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.8	3.1
c35	18	18	-	-	-	18	-	-	-	18	-	-	-	0.1	1.6	2.8
c36	18	18	-	-	-	18	-	-	-	18	-	-	-	0.1	1.2	2.3
c37	18	18	-	-	-	16	1.8	2.4	2.1	11	1.5	3.9	2.6	1.3	0.0	0.7
c38	18	10	1.4	22.3	8.9	11	1.3	8.6	3.3	9	2.0	9.4	4.5	1.3	2.3	13.5
c39	18	15	2.9	8.8	5.6	16	1.6	4.0	2.8	9	1.2	3.7	2.5	1.6	0.6	5.5
c40	18	9	11.8	23.6	17.9	9	3.0	8.8	4.8	9	4.2	10.3	6.6	2.4	5.7	14.5
c41	18	18	-	-	-	18	-	-	-	18	-	-	-	0.2	1.5	2.9
c42	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.3	1.9
c43	18	18	-	-	-	18	-	-	-	18	-	-	-	0.2	1.8	2.7
c44	18	18	-	-	-	18	-	-	-	18	-	-	-	-0.1	0.9	1.2
c45	18	18	-	-	-	18	-	-	-	18	-	-	-	1.6	0.5	0.8
c46	18	14	1.8	6.8	4.1	14	1.2	2.6	1.7	11	1.6	3.2	2.0	1.4	0.8	3.9
c47	18	18	-	-	-	18	-	-	-	18	-	-	-	1.6	0.6	0.8
c48	18	13	4.6	16.5	9.6	12	1.9	3.5	2.9	9	1.2	7.7	3.0	1.7	1.9	10.1
c49	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.5	2.8
c50	18	18	-	-	-	18	-	-	-	18	-	-	-	1.2	1.1	2.1
c51	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.2	1.8
c52	18	18	-	-	-	18	-	-	-	18	-	-	-	1.0	1.2	2.0
c53	18	18	-	-	-	16	1.0	1.1	1.1	9	1.3	1.6	1.4	1.6	0.6	0.8
c54	18	10	1.7	5.5	3.4	10	1.8	3.1	2.4	9	2.7	4.1	3.5	1.3	0.6	2.6
c55	18	18	-	-	-	17	1.0	1.0	1.0	9	1.4	1.7	1.5	1.7	0.8	1.1
c56	18	13	2.3	4.3	3.4	10	1.2	3.5	1.9	9	2.3	4.5	3.2	1.5	0.5	2.3
c57	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.1	1.8
c58	18	18	-	-	-	18	-	-	-	18	-	-	-	0.7	0.9	1.6
c59	18	18	-	-	-	18	-	-	-	18	-	-	-	0.5	1.3	1.9
c60	18	18	-	-	-	18	-	-	-	18	-	-	-	0.7	1.3	2.3
c61	18	18	-	-	-	17	1.0	1.0	1.0	10	1.0	1.3	1.2	1.3	0.6	0.8
c62	18	9	1.9	8.0	5.4	10	1.7	3.4	2.8	9	3.6	5.2	4.3	1.5	1.2	3.4
c63	18	18	-	-	-	18	-	-	-	12	1.1	1.4	1.2	1.5	0.8	1.0
c64	18	11	1.8	5.2	3.4	9	1.1	2.8	2.0	9	2.2	4.5	3.1	1.4	0.7	2.5

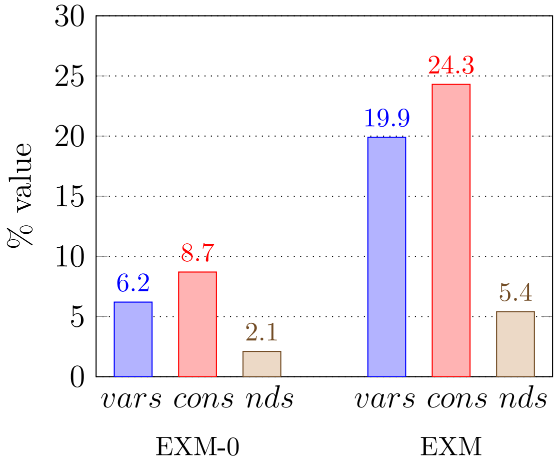
Note:"-" represents that all instances in the corresponding instance set are solved to optimality by the corresponding method.

Table 2.5 gives a different view of the results of Table 2.4 by grouping the instances based on the 31 "C" classes. In the table, column "*ni*" gives the number of instances in the corresponding group and "*opt*" is the number of instances solved to optimality.

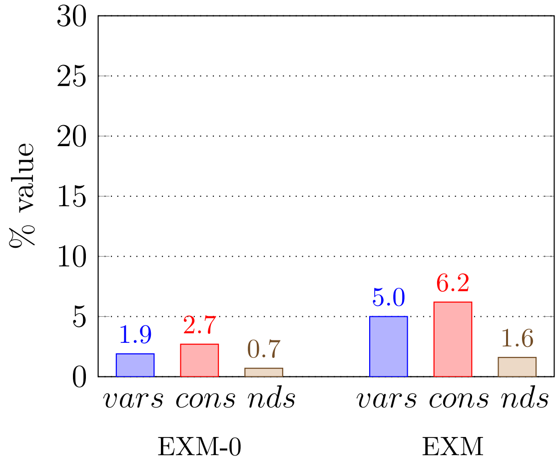
Table 2.4 indicates that EXM achieves a similar performance to that of EXM-0 when used to solve the CTSNDP. The detailed Table 2.5 shows that EXM solved 491 instances to optimality, whereas 500 instances were solved by EXM-0 but that there are instances solved to optimality by EXM that cannot be solved by EXM-0 within the imposed time limit, and vice versa. Interestingly, on the one hand EXM requires (on average) a smaller number of iterations to converge to an optimal solution than EXM-0, thus showing the importance of the refinement strategy in DDD algorithms. On the other hand, EXM requires a higher computing time, as shown by column $\%tLB$. For both EXM and EXM-0, most of the computing time is taken up by the MIP solver used to compute the lower bounds at the different iterations. These results show that tuning a DDD algorithm requires finding the right trade-off between the time spent by the MIP solver and the rate of convergence of the DDD method, which strictly depends on the refinement strategy.

The results on the CTSNDP-HC instances show that CTSNDP-HC instances of groups **LC/LF** and **LC/HF** can also be easily solved by EXM, as with the CTSNDP case. Conversely, the instances of groups **HC/LF** and **HC/HF** are more difficult to solve, as shown by the percentages of instances solved to optimality. In particular, for groups **HC/HF** that are characterized by a high cost ratio and high flexibility, a trade-off between fixed, flow and holding costs must be achieved.

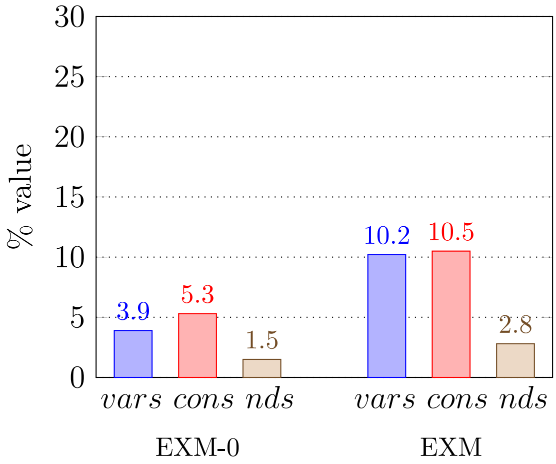
The summary results of Table 2.4 and the detailed results of Table 2.5 show that, compared with EXM-0, algorithm EXM generates better lower and upper bounds for the CTSNDP-HC. On the most difficult instances of group **HC/HF**, the solutions obtained by EXM improve those derived from the solutions obtained by EXM-0 significantly. This is indicated by the maximum percentage deviation $\%UB1$ being equal to 14.5%. It thus implies that a significant cost saving can be gained in some cases by taking into



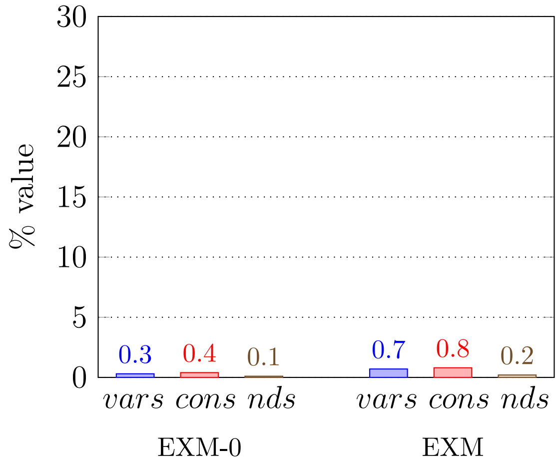
(a) Group HC/LF



(b) Group HC/HF



(c) Group LC/LF



(d) Group LC/HF

Figure 2.6 Comparison of partially and fully time-expanded networks

account holding costs when solving the CTSNDP. It is worth noting that the negative $\%LB0$ value in Table 2.5 is due to applying the optimally gap of 0.01.

To further analyze the effectiveness of the DDD approach and impact of the holding costs, Figure 2.6 shows the relative sizes (average percentage values), in terms of the number of variables ("*vars*") and constraints ("*cons*") of the final relaxation models (models SND($\mathcal{D}_\mathcal{T}$) and SND-HC-R($\mathcal{D}_\mathcal{T}$) associated with the last iteration of the DDD algorithm) solved by algorithms EXM-0 and EXM with respect to the models associated with the fully time-expanded networks. The figure also shows the relative number of time points (or nodes) ("$\%nds$") of the final partially time-expanded network $\mathcal{D}_\mathcal{T}$

Table 2.6 CTSNDP-HC instances: holding costs, holding times and consolidations

	HC/LF			**HC/HF**			**LC/LF**			**LC/HF**		
	%hc	%ht	%cs	%hc	%ht	%cs	%hc	%ht	%cs	%hc	%ht	%cs
UB1	1.9	10.3	47.7	2.3	13.7	59.9	1.8	7.7	14.5	1.8	7.3	9.3
UB	1.2	7.0	47.5	1.7	10.4	61.3	0.5	2.3	13.0	0.3	1.5	8.0

over the fully time-expanded network. The figure clearly shows the advantage of the DDD approach, which is capable of computing optimal solutions considering only a reduced set of time points of the fully time-expanded network. What is more, the sizes of the MIP solvers solved at the different iterations are confined to small portions of the model associated with the fully time-expanded networks, a very relevant feature given the complexity of solving TI formulations. A comparison with the results using EXM-0 shows that EXM requires about twice the number of variables, constraints and time points of EXM-0, which indicates an increased complexity of the problem after incorporating holding costs.

Moreover, Table 2.6 summarizes relevant details of the solutions corresponding to the upper bounds $UB1$ and UB. More specifically, the table reports the following average percentage values: (i) $\%hc$, the total holding cost over the total solution cost, (ii) $\%ht$, the total holding time over the total transit time, and (iii) $\%cs$, the number of consolidation arcs over the total number of arcs used by the solution. The table also shows that the solutions computed by EXM achieve a marginal reduction in terms of holding times (and corresponding holding costs) with respect to the solutions derived from the EXM-0. Interestingly, this reduction is achieved by increasing the percentages of consolidations (see column $\%cs$) for group **HC/HF**, and by decreasing of the percentages for the remaining groups. These also reveal the significant effect of the holding costs on the decisions of holding and consolidation for solving the CTSND

2.4.2 Experiments on Newly Generated CTSNDP-HC Benchmark Instances

For our second set of experiments, we generated a new set of CTSNDP-HC instances. To investigate attributes of the instances that affect the complexity of the CTSNDP-HC and the importance of the holding costs, we exploit two main components characterizing service network design problems: the connectivity of the underlying physical network (spatial component) and the flexibility of the shipments' time requirements (temporal component). Using these newly generated instance, we also examine the benefits gained by incorporating holding costs into the CTSNDP and the impact of the holding costs on the solution structure in these more differentiated instances.

Instance Generation

The procedure used to generate the new instances followed two main steps.

(1) **Varying the connectivity level**. For each instance of Table 2.2, based on the corresponding network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, we first derive the travel times τ_{ij}, $(i, j) \in \mathcal{A}$ using the method proposed by Boland et al. [20]. Similarly to the generation of the instances explained in Section 2.4.1, for each $k \in \mathcal{K}$, we set the per-unit-of-demand-and-time cost h_i^k equal to $0.3\,\epsilon_i$ for $i \in \mathcal{N}$ with $\epsilon_i = 1/|\mathcal{A}_i| \sum_{a \in A_i}(c_a + f_a/u_a)/\tau_a$, and set $h_{d^k}^k = 0$. Let Γ be the time of the path in $\mathcal{D}$ having maximum time among the shortest-time $o^k - d^k$ paths associated with the vertices o^k, d^k, $\forall k \in \mathcal{K}$. Then, we reduce the number of arcs of the network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ by an arc reduction procedure that at each iteration performs the following steps:

   (i) Randomly select an arc α by means of a uniform distribution from the set of arcs $\mathcal{A}$.

   (ii) Check the connectivity of the network $(\mathcal{N}, \mathcal{A} \setminus \{\alpha\})$, i.e., check if for every pair of vertices o^k, d^k, $k \in \mathcal{K}$, there exists a path connecting o^k to d^k.

   (iii) If the graph is connected, set $\mathcal{A} = \mathcal{A} \setminus \{\alpha\}$ and begin a new iteration.

If the removal of an arc α results in a disconnected network, a new arc is randomly selected and the procedure terminates after $\frac{1}{10}x$ unsuccessful removal attempts, where x is the initial number of arcs. After termination, each remaining arc in the resulting graph $\mathcal{D}$ is in turn selected and tested for removal.

Let NR be the total number of arcs removed. We consider four final networks corresponding to the networks obtained after the removal of $\lfloor xNR \rfloor$ arcs where $x \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$, denoted as $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$, respectively.

(2) **Varying the flexibility level**. Given a network $\mathcal{D}_x$, $x = 1, 2, 3, 4$, let $\underline{\tau}_{ij}$, $i, j \in \mathcal{N}$, $i \neq j$ be the length of the least total travel time path from i to j, and let $\mathcal{B} = \{(i, j) : \underline{\tau}_{ij} \leq \Gamma\}$.

If, for a commodity $k \in \mathcal{K}$, we have $\underline{\tau}_{o^k d^k} > \Gamma$, we assign to the commodity new origin and destination nodes by randomly sampling with a uniform distribution a new pair from set $\mathcal{B}$, and we recompute the new value $\underline{\tau}_{o^k d^k}$.

We then generate available and due times also based on the method proposed by Boland et al. [20] as follows:

(a) We compute the average length computed as $l_{avg} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \underline{\tau}_{o^k d^k}$.

(b) For generating values e^k, we create a normal distribution with mean l_{avg} and standard deviation $\frac{1}{6} l_{avg}$.

(c) For generating values l^k, we create three normal distributions (denoted as A, B and C, respectively) from which we drawn values l^k, all of which are defined by a standard deviation $\frac{1}{6}\mu$ but where we consider the values for the mean μ, $\frac{1}{2} l_{avg}$, l_{avg} and $\frac{3}{2} l_{avg}$. A value l^k is set equal to $e^k + \underline{\tau}_{o^k d^k} + \mathscr{F}_k$ where $\mathscr{F}_k \geq 0$ is the value drawn from a distribution.

Based on the above two steps, for each of the instances in Table 2.2, we generated four different networks $\mathcal{D}_x$, $x = 1, 2, 3, 4$, and three instances based on the three different distributions for values e^k and l^k, $k \in \mathcal{K}$. These steps were repeated three times to finally obtain a total of $3 \times (31 \times 4 \times 3) = 1116$ instances.

Table 2.7 Connectivity properties of the new CTSNDP-HC instances

Network	Min cut.	Avg cut.
$\mathcal{D}_1$	7	12
$\mathcal{D}_2$	4	8
$\mathcal{D}_3$	2	4
$\mathcal{D}_4$	1	1

Table 2.8 Time flexibility of the new CTSNDP-HC instances

Normal Distribution	Mean(μ)	StdDev(σ)
A	$\frac{1}{2}l_{avg}$	$\frac{1}{6}\mu$
B	l_{avg}	$\frac{1}{6}\mu$
C	$\frac{3}{2}l_{avg}$	$\frac{1}{6}\mu$

Table 2.7 summarizes the connectivity properties of the instances generated. Given an instance and the associated network $\mathcal{D}_x$, $x \in \{1, 2, 3, 4\}$, we first computed the minimum $o^k - d^k$ cut (denoted as $(S, \mathcal{N} \setminus S)_k$) in $\mathcal{D}_x$, $\forall k \in \mathcal{K}$, and then we computed $\min_{k\in\mathcal{K}}\{|(S, \mathcal{N} \setminus S)_k|\}$, i.e., the cardinality of the cut having the minimum cardinality among the different $o^k - d^k$ pair. For each type of network, the table reports the cardinality of the cut having the minimum cardinality ("Min cut") and the average cardinality of all the minimum $o^k - d^k$ cuts ("Avg cut") computed over all instances belonging to this particular type of network. Norms and standard deviations of three normal distributions (distribution A, B and C) are summarized in Table 2.8.

Results

For each of the newly generated CTSNDP-HC instances, we excused algorithm EXM and EXM-0, with setting the holding costs equal to zero for EXM-0. For these experiments, we used $tol_{DDD} = 0.01$ and a time limit of two hours for both EXM and EXM-0, and $tol_{MIP} = 0.01$ for EXM-0, whereas for EXM, the tol_{MIP} was computed dynamically as $\max\{\min\{0.04, gap \times 0.25\}, 0.01\}$.

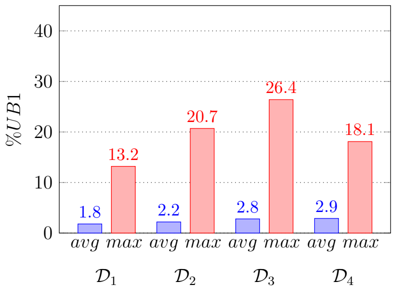
Table 2.9 summarizes the results obtained using the same notation introduced in Table 2.4. The instances were grouped by distribution and network type. The average values of *time*, and *iter* were computed over all instances. The deviations relative to $UB0$ and UB were computed over all instances not solved to optimality, whereas the deviations relative to $LB0$ were computed over all instances.

Table 2.9 Summary results on the new CTSNDP-HC instances

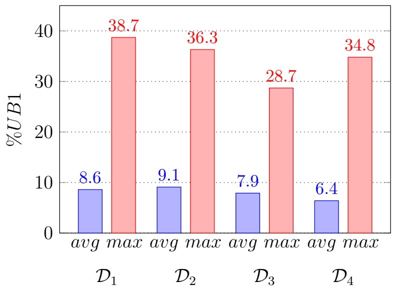
		Zero holding costs							Nonzero holding costs							
		EXM-0							EXM							
			$\%UB0$							$\%UB$						$\%LB0$
Dist.	Network	$\%opt$	min	max	avg	$time$	$\%tLB$	$iter$	$\%opt$	min	max	avg	$time$	$\%tLB$	$iter$	avg
A	$\mathcal{D}_1$	86.0	1.1	18.1	7.0	1372.2	81.1	6.9	84.9	1.0	7.0	2.7	1577.8	88.2	7.5	2.0
	$\mathcal{D}_2$	86.0	1.6	25.1	9.7	1448.9	79.5	5.9	76.3	1.0	6.0	2.6	1919.7	86.2	6.8	2.2
	$\mathcal{D}_3$	82.8	1.1	30.2	9.1	1541.0	77.7	5.9	79.6	1.0	3.5	2.0	1853.5	83.6	7.3	2.5
	$\mathcal{D}_4$	86.0	1.4	24.2	10.4	1176.2	76.5	6.0	84.9	1.1	6.3	2.5	1456.4	82.7	7.9	2.7
B	$\mathcal{D}_1$	55.9	1.4	46.7	19.1	4389.6	89.1	3.6	55.9	1.0	14.2	4.7	3384.9	93.9	6.3	3.7
	$\mathcal{D}_2$	53.8	1.1	47.5	18.1	4535.6	86.2	3.5	57.0	1.2	13.1	4.5	3416.6	92.2	6.2	3.9
	$\mathcal{D}_3$	55.9	2.6	37.0	14.4	4418.3	80.4	3.0	53.8	1.0	17.9	4.2	3596.4	88.1	6.6	3.5
	$\mathcal{D}_4$	74.2	2.6	41.8	13.5	2650.3	78.1	3.9	67.7	1.0	13.7	3.7	2719.4	85.0	8.0	3.2
C	$\mathcal{D}_1$	47.3	1.1	54.7	21.6	3892.0	93.0	3.0	45.2	1.4	40.1	8.1	3986.3	97.2	5.5	4.0
	$\mathcal{D}_2$	48.4	1.3	57.1	19.9	3769.7	90.6	2.8	46.2	1.2	45.3	7.9	3976.2	95.9	5.7	3.9
	$\mathcal{D}_3$	51.6	2.0	39.8	15.2	3651.2	86.3	3.0	47.3	1.0	19.8	5.7	3859.0	93.5	5.9	3.7
	$\mathcal{D}_4$	66.7	1.3	42.8	12.6	2592.8	80.8	3.7	62.4	1.1	15.8	4.5	3118.5	88.4	8.0	3.3

We recall that an increasing time flexibility corresponds to the ordering of distributions A, B and C, whereas a decreasing connectivity level is associated with the ordering of networks $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$. The results shown in the table indicate that, for both EXM-0 (CTSNDP case) and EXM (CTSNDP-HC case), the instances characterized by high flexibility and connectivity levels are particularly difficult. Such difficulty is due to the following reason: Under higher time flexibility and connectivity, more consolidation opportunities are allowed, as well as more alternative services are available. These increase the difficulty of the MIP problems to be solved, displaying many equivalent solutions that significantly blows up the sizes of the branch-and-bound trees. As a consequence, and as also shown by the percentage of the time spent by the MIP solver and by the number of iterations, the entire solution process is slowed down.

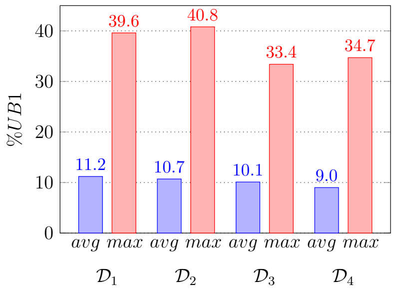
Table 2.9 also reveals a significant percentage improvements of the lower bound LB obtained by EXM respect to the lower bound $LB0$ provided by EXM-0, i.e., up to 4%. Moreover, Figure 2.7 plots the average and maximum percentage deviations of the upper bound $UB1$ over all the instances of each distribution and network type. It indicates that the cost saving percentage of the minimal total cost achieved by EXM is also significant, whereas its maximum value can be up to 40.8%, and its average value grows with time flexibility. The percentage deviations with respect to the lower bound $LB0$ and of the upper bound $UB1$ clearly show the effectiveness of the EXM in



(a) Distribution A



(b) Distribution B



(c) Distribution C

Figure 2.7 Results about upper bound $UB1$ showing the average and maximum cost saving percentages gained by the CTSNDP-HC solutions over CTSNDP-based solutions

solving the CTSNDP-HC, and indicate that solving the CTSNDP and, based on the corresponding solutions, deriving corresponding CTSNDP-HC solutions, is clearly not a valid option.

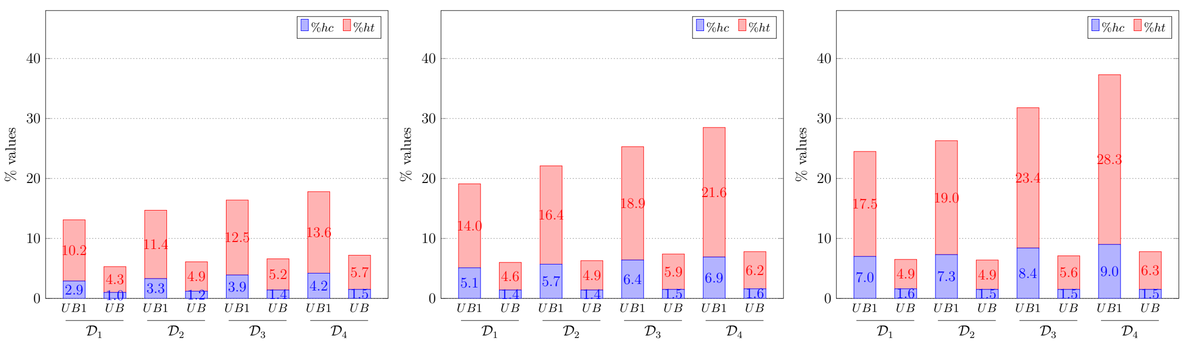
Since the newly generated CTSNDP-HC instances are based on the 31 “C" classes of the CTSNDP benchmark instances, they can also be classified into 31 groups. In the analysis below, we only consider the CTSNDP-HC instances from the groups in which all instances are solved to optimality by both EXM and EXM-0.

Table 2.10 presents the relative sizes of the partially time-expanded networks over the fully time-expanded networks of EXM-0 and EXM for the newly generated CTSNDP-HC instances mentioned above, where the results were grouped by distribution and network type and the notation adopted is that of Figure 2.6. In addition, the table also shows percentage values concerning the number of consolidations (“$\%cs$") associated with upper bounds $UB1$ and UB. Finally, Figure 2.8 depicts the ratios of the holding costs and holding times of these considered instances.

The results concerning the percentages of the number of consolidations (column “$\%cs$”) given by the table and the percentages of the holding times and costs depicted by the figure, show that significant reductions in the two measures are achieved by upper bound UB with respect to upper bound $UB1$, again depending on the connectivity of the underlying physical network and the flexibility of the shipments’ time requirements. This confirms the significant impact of the holding costs on the decisions of holding and

Table 2.10 Sizes of the partially and fully time-expanded networks and consolidations

Dist.	Network		$\%vars$	$\%cons$	$\%nds$	Upp. bounds	$\%cs$
A	$\mathcal{D}_1$	EXM-0	1.7	2.3	0.8	$UB1$	40.7
		EXM	3.8	4.3	1.4	UB	38.0
	$\mathcal{D}_2$	EXM-0	1.6	2.1	0.7	$UB1$	42.9
		EXM	3.4	3.9	1.3	UB	41.1
	$\mathcal{D}_3$	EXM-0	1.4	1.9	0.7	$UB1$	49.6
		EXM	2.9	3.3	1.1	UB	43.8
	$\mathcal{D}_4$	EXM-0	1.3	1.8	0.7	$UB1$	51.0
		EXM	2.5	3.2	1.2	UB	47.3
B	$\mathcal{D}_1$	EXM-0	1.0	1.4	0.6	$UB1$	49.4
		EXM	2.4	3.1	1.1	UB	41.3
	$\mathcal{D}_2$	EXM-0	0.9	1.3	0.6	$UB1$	53.3
		EXM	2.2	2.7	1.0	UB	42.2
	$\mathcal{D}_3$	EXM-0	0.8	1.1	0.5	$UB1$	55.4
		EXM	2.0	2.4	0.9	UB	46.5
	$\mathcal{D}_4$	EXM-0	0.7	1.0	0.6	$UB1$	57.8
		EXM	1.7	2.1	1.0	UB	47.8
C	$\mathcal{D}_1$	EXM-0	0.7	1.1	0.5	$UB1$	49.9
		EXM	1.8	2.5	1.0	UB	40.9
	$\mathcal{D}_2$	EXM-0	0.7	1.0	0.5	$UB1$	56.0
		EXM	1.8	2.4	0.9	UB	42.3
	$\mathcal{D}_3$	EXM-0	0.7	1.0	0.4	$UB1$	58.3
		EXM	1.7	2.2	0.9	UB	42.0
	$\mathcal{D}_4$	EXM-0	0.5	0.7	0.4	$UB1$	60.1
		EXM	1.5	1.8	0.8	UB	47.2



(a) Distribution A (b) Distribution B (c) Distribution C

Figure 2.8 Ratios of the holding costs and holding times

Table 2.11 Analysis of the difference in terms of timed arcs used between upper bounds $UB1$ and UB

Dist.	Network	$\%da$	Dist.	Network	$\%da$	Dist.	Network	$\%da$
A	$\mathcal{D}_1$	8.66%	B	$\mathcal{D}_1$	9.51%	C	$\mathcal{D}_1$	11.55%
	$\mathcal{D}_2$	9.26%		$\mathcal{D}_1$	9.51%		$\mathcal{D}_2$	11.87%
	$\mathcal{D}_3$	8.77%		$\mathcal{D}_3$	11.05%		$\mathcal{D}_3$	12.31%
	$\mathcal{D}_4$	9.46%		$\mathcal{D}_4$	13.90%		$\mathcal{D}_4$	15.74%

consolidation for solving the CTSNDP. Moreover, due to the reduction in the number of consolidations, solutions obtained by EXM are more reliable than solutions obtained by EXM-0. Indeed, in practice, routing issues that may occur along a trip often cause delays to subsequent consolidations, which result in late services and deliveries. As observed for the first set of instances, Table 2.10 also confirms the effectiveness of the DDD approach in confining the sizes of the different MIPs to small portions of the fully time-expanded model. In order to analyze the differences between the paths used in the solutions of upper bounds UB and $UB1$, for each instance and for each commodity $k \in \mathcal{K}$, we computed the ratio between the number of different timed arcs used by the paths of commodity k of solutions UB and U$B1$ and the total number of timed arcs used by the solutions, i.e., we computed the ratio

$$r_k = \frac{|A(UB1) \cup A(UB)| - |A(UB1) \cap A(UB)|}{|A(UB1) \cup A(UB)|},$$

where $A(UB)$ and $A(UB1)$ are the sets of timed arcs used in the solutions of UB and $UB1$, respectively. Then, for the given instance, the percentage average of values r_k were computed as $\%da = \frac{\sum_{k \in \mathcal{K}} r_k}{|\mathcal{K}|}$. Table 2.11 reports average values $\%da$ computed for the different instances and grouped by distribution and network types. The table shows that on average the solutions differ for about 11% of the total timed arcs used. The differences are more significant for increasing flexibility level, and for a fixed flexibility level, for decreasing connectivity level. The results reveal the ineffectiveness of the deriving method of the CTSNDP solution proposed in Boland et al. [20] in handling non-zero holding costs cases, and also demonstrate the significant impact of the holding

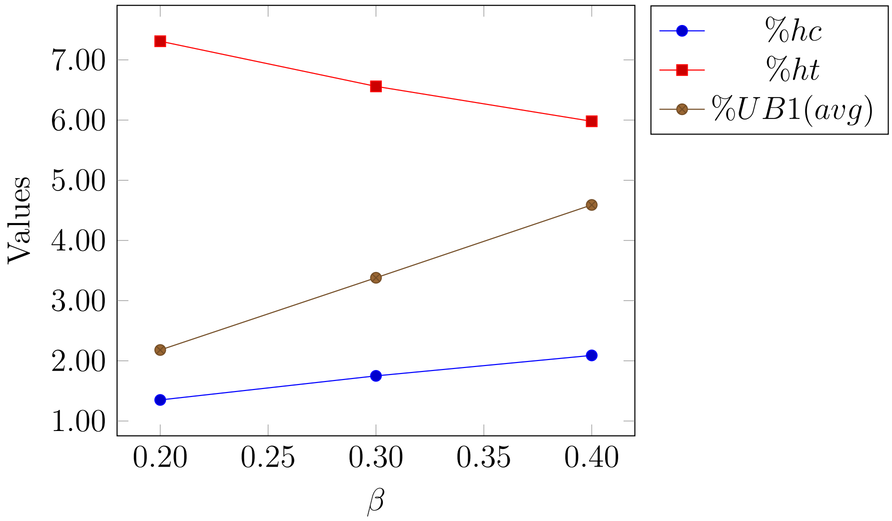


Figure 2.9 Sensitive analysis on the per-unit-of-demand-and-time cost h_i^k

costs on the solution structure, highlighting the importance of considering the holding costs in solving the CTSNDP-HC, especially for instances with high flexibility and lower connectivity.

To further analyze the effect of varying the per-unit-of-demand-and-time cost h_i^k on the CTSNDP-HC that was set equal to $\beta\,\epsilon_i$, $\forall k \in \mathcal{K}$, with $\beta = 0.3$ (see Section 2.4.1) in our experiments, we compare the results obtained by EXM for the case $\beta = 0.3$ with the results obtained using $\beta = 0.2$ and $\beta = 0.4$. For the experiments, we considered a restricted set of instances composed of the instances of networks $\mathcal{D}_2$ and $\mathcal{D}_3$ under the three different distributions A, B, and C. Figure 2.9 summarizes the results obtained. For each β value (x axis), the figure shows the following average percentage values: (i) $\%hc$, the total holding cost over the total solution cost, (ii) $\%ht$, the total holding time over the total transit time, and (iii) $\%UB1_{avg}$, the average percentage deviation of upper bound $UB1$ with respect to upper bound UB. The percentage values were computed over all instances solved to optimality with all the considered β values. The figure shows that when β increases, both $\%UB1_{avg}$ and $\%hc$ increase. This implies that although a higher per-unit-of-demand-and-time holding cost induces a larger percentage of the holding cost, it induces a more significantly cost saving achieved by UB over $UB1$. From the figure, we can also observe that when β increases, $\%ht$

decreases. This implies that a higher per-unit-of-demand-and-time holding cost also induces a significant reduction in holding time. Hence, as per-unit-of-demand-and-time holding cost increases, it is more beneficial to take into account holding costs for solving the CTSNDP-HC.

2.5 Summary

In this chapter, we designed a new exact algorithm for a generalization of the continuous-time service network design problem (CTSNDP), first studied by Boland et al. [20], where in addition to fixed and flow costs, holding costs are also considered (CTSNDP-HC). We proved the importance of incorporating holding costs into the CTSNDP by showing that, in some situations, the cost saving percentage of the minimal total cost can be significantly large. The exact algorithm uses the same dynamic discretization discovery (DDD) solution framework proposed by Boland et al. [20] for the CTSNDP, but it extends the DDD framework in a number of non-trivial ways by exploiting a new relaxation of a complete time-index model, a new upper bound heuristic and a new refinement strategy. The new algorithm was extensively tested both on instances derived from the literature and on newly generated instances, with the aim of benchmarking the essential factors of the CTSNDP-HC. In particular, to assess the impact of the holding costs, we designed experiments by varying the connectivity of the underlying physical network and the flexibility of the shipments' time requirements. The results obtained not only showed the effectiveness of the new exact algorithm in solving challenging CTSNDP-HC instances involving up to 400 commodities, but also indicated that ignoring the holding costs leads to poor quality solutions. The impact of holding costs is significant particularly when the network is characterized by high flexibility and low connectivity levels. Compared with the heuristic solution obtained by the method ignoring the holding costs, the maximum cost saving percentage of the minimal total cost achieved by our exact method can be up to 40.8%.

Chapter 3

Robust Continuous-Time Service Network Design with Uncertain Travel Times

3.1 Introduction

In Chapter 2, we incorporated the holding costs in the continuous-time service network design, and it resulted in the CTSNDP-HC, for which we proposed the first exact optimization method, and where the problem instances are all deterministic. This chapter aims to further incorporate uncertain travel times into the continuous-time service network design, for which we propose the first robust optimization model and corresponding solution methods for the CTSNDP-HC under travel time uncertainty.

In deterministic service network design, problem parameters, such as demands and travel times, are assumed to be known in advance and unchanged during the planning horizon. However, this is not always the case in many practical situations, where decision-makers face various uncertainties. Ignoring such uncertainties often leads to poor and undesirable decisions. Hence, in such situations, uncertainties need to be taken into account in the planning process. There are two main sources of uncertainties in the SNDP, i.e., demands and travel times. However, most of the existing studies that incorporated uncertainties in the SNDP focused only on demand uncertainty (see, for example, [79, 66, 6, 108, 5, 73]). Despite its importance, travel time uncertainty has seldom been taken into account in existing studies of the SNDP. Our study presented here is the first one that incorporates travel time uncertainty in solving the CTSNDP-HC.

In Chapter 2, we proposed a TI model based on a time-expanded network for the deterministic CTSNDP-HC. However, this TI model cannot be extended to derive a robust optimization model for the CTSNDP-HC under travel time uncertainty. This is because that in order to model the travel time uncertainty, the complete time-expanded network needs to contain service arcs for all possible travel times and contain time points for all possible arrival and departure times. This makes the complete TI model significantly more complicated than that for the deterministic CTSNDP-HC, being intractable to solve. As a result, common solution methods, such as the dynamic discretization discovery method, the Benders decomposition method and the column generation method, cannot be effectively designed based on such a complicated complete TI model for the CTSNDP-HC under travel time uncertainty. Therefore, the TI formulation proposed in Chapter 2 cannot be utilized and extended to develop a tractable robust optimization model for the CTSNDP-HC under travel time uncertainty.

To tackle this challenge, instead of using the TI model which contains time-index variables and constraints based on the time-expanded network, we index the consolidations on each service arc to derive a new formulation of the deterministic CTSNDP-HC based on the flat network. By extending this new formulation of the deterministic CTSNDP-HC, we can then derive a two-stage robust optimization model for the CTSNDP-HC under travel time uncertainty, where a budgeted uncertainty set is used to formulate the travel time uncertainty. The first stage of the robust optimization model determines the selection of services, and the routing and consolidation plans of commodities. The second stage determines the departure schedule after the actual values of the travel times are realized.

To solve our newly proposed two-stage robust optimization model, we first develop a standard column-and-constraint generation (C&CG) method. It is based on a novel reformulation of the second-stage problem of the model, and adapts the solution framework proposed for the general two-stage robust optimization model in Zeng and Zhao [111]. We then introduce a parameter to control the number of consolidations on each arc, and enhance the standard C&CG method by dynamically adjusting the value of this parameter via some new optimization techniques. Extensive computational results indicate that the enhanced C&CG method significantly outperforms the standard C&CG method, in terms of shorter computational time and better solution quality. The computational results also reveal that the dynamic parameter adjustment is effective in accelerating the C&CG method. Moreover, we evaluate the robustness of solutions obtained from the newly proposed two-stage robust optimization model, and demonstrate the value of incorporating travel time uncertainty in the CTSNDP-HC as well as the price of the robustness against travel time uncertainty.

The remainder of this chapter is organized as follows. In Section 3.2, we extend the setting of the deterministic CTSNDP-HC to introduce the definition of the robust CTSNDP-HC under travel time uncertainty. In Section 3.3, we present a new formulation of the deterministic CTSNDP-HC, based on which, we derive a two-stage robust optimization model for the CTSNDP-HC under travel time uncertainty. We then present the standard column-and-constraint generation method in Section 3.4, and elaborate the enhanced C&CG method via dynamic parameter adjustment in Section 3.5, to solve the proposed two-stage robust optimization model. Two additional acceleration strategies for the solutions methods are illustrated in Section 3.6. The experimental results are reported in Section 3.7 followed by a summary of this chapter in Section 3.8.

3.2 Problem Descriptions

In this section, we first reexamine the definition of the deterministic CTSNDP-HC under given travel times, where feasible solutions are defined based on the flat network instead of the time-expanded network. Based on this, we then introduce and formulate a robust CTSNDP-HC under travel time uncertainty.

3.2.1 Deterministic CTSNDP-HC

In Chapter 2, we extended the problem setting in Boland et al. [20] to define the deterministic CTSNDP with holding costs incorporated, and we defined its feasible solutions based on the time-expanded network. In this section, we reexamine the definition of the deterministic CTSNDP-HC where feasible solutions are defined based on the flat network, 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*. Consider a commodity set $\mathcal{K}$, where each commodity $k \in \mathcal{K}$ has a single origin $o^k \in \mathcal{N}$, a single destination $d^k \in \mathcal{N}$, a transportation quantity $q^k \in \mathbb{N}_{>0}$, an earliest available time $e^k \in \mathbb{N}$ for its departure from the origin o^k, and a due time $l^k \in \mathbb{N}_{>0}$ for its arrival at the destination d^k. The transportation quantity cannot be split so that it must be picked up only once from the origin after the earliest available time and delivered to the destination before the due time along a single delivery path.

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 cost $c^k_{i,j} \in \mathbb{R}_{>0}$ for each commodity $k \in \mathcal{K}$; (3) a fixed (resource installation) cost $f_{i,j} \in \mathbb{R}_{>0}$ for service on the arc; and (4) a capacity $u_{i,j} \in \mathbb{N}_{>0}$ for service on the arc. Besides, both *in-transit* and *in-storage* holding costs are considered here. The in-transit holding costs are incorporated in the flow costs $c^k_{i,j}$ for $k \in \mathcal{K}$ and $(i, j) \in \mathcal{A}$. A unit in-storage holding cost $h^k_i \in \mathbb{R}_{\geq 0}$ (per unit of flow and unit of time) is incurred, when a commodity $k \in \mathcal{K}$ is stored at node i for $i \in \mathcal{N}$.

The deterministic CTSNDP-HC needs to decide delivery paths and departure times for all the commodities as well as their consolidation plan, so as to determine the installation of resources for services required to transport these commodities. Its aim is to satisfy the delivery time constraints imposed by the commodities' earliest available times and due times, with the total cost, including fixed costs, flow costs and holding costs, minimized.

A *feasible solution* to the deterministic CTSNDP-HC consists of (i) a routing plan, (ii) a consolidation plan, and (iii) a departure schedule, which are defined below. We call a directed path P in the flat network $\mathcal{D}$ as 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 practice, the delivery path of each commodity cannot have repeated vertices or arcs, and thus, it must be an elementary flat path from the origin to the destination of the commodity. Accordingly, a *routing plan* $\mathcal{P}$ consists 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 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 $\mathcal{P}$, we need to specify how shipments of the commodities are consolidated for each arc of the flat network $\mathcal{D}$. For each $\alpha \in \mathcal{A}$, let $\mathcal{K}(\alpha) = \{k \in \mathcal{K} \mid a_n^k = \alpha$ for some n with $1 \leq n \leq m^k\}$ indicate the subset of commodities that pass through arc α of their flat paths in $\mathcal{P}$. We can then represent a consolidation on arc α by a commodity subset $C \subseteq \mathcal{K}(\alpha)$, so that shipments of commodities $k \in C$ on arc α are consolidated to be shipped together. Since the demand quantity of each commodity cannot be split in delivery, there are at most $|\mathcal{K}|$ consolidations on each arc. Accordingly, a *consolidation plan* $\mathcal{C}$ consists of consolidations $C_1^\alpha, C_2^\alpha, \ldots, C_{|\mathcal{K}|}^\alpha$ for $\alpha \in \mathcal{A}$, with each $C_r^\alpha \subseteq \mathcal{K}(\alpha)$ for $r = 1, 2, \ldots, |\mathcal{K}|$, indicating the r-th consolidation on arc α. Each consolidation C_r^α, for $\alpha \in \mathcal{A}$, $r = 1, 2, \ldots, |\mathcal{K}|$ can be empty. We refer to the subscript r as the *consolidation index* of consolidation C_r^α on arc α. If consolidations C_r^α for $r = 1, 2, \ldots, |\mathcal{K}|$ cover all $k \in \mathcal{K}(\alpha)$ for each $\alpha \in \mathcal{A}$, i.e.,

$$\bigcup_{r=1}^{|\mathcal{K}|} C_r^\alpha = \mathcal{K}(\alpha), \text{ for each } \alpha \in \mathcal{A},$$

then we call such a $(\mathcal{P}, \mathcal{C})$ pair a *flat solution* to the deterministic CTSNDP-HC.

Given a flat solution $(\mathcal{P}, \mathcal{C})$, we need to further specify when each commodity departs from every node that it passes. 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* $\mathcal{T}$ consists of departure times $t^k_{\nu^k_n}$ for $k \in \mathcal{K}$ and $n \in \{1, 2, \ldots, m^k\}$, indicating the departure time when commodity k departs from node ν^k_n of its flat path P^k. We can now define that $(\mathcal{P}, \mathcal{C}, \mathcal{T})$ forms a *feasible solution* to the deterministic CTSNDP-HC if the departure schedule $\mathcal{T}$ satisfies that

$$t^k_{\nu^k_n} \geq e^k, \quad \text{for } n = 1, \tag{3.1}$$

$$t^k_{\nu^k_{n+1}} \geq t^k_{\nu^k_n} + \tau_{a^k_n}, \text{ for } n \in \{1, 2, \ldots, m^k - 1\}, \tag{3.2}$$

$$t^k_{\nu^k_n} + \tau_{a^k_n} \leq l^k, \text{ for } n = m^k, \tag{3.3}$$

$$t^k_i = t^{k'}_i, \quad \text{for } k \in C_r^{(i,j)} \text{ and } k' \in C_r^{(i,j)} \text{ with } (i,j) \in \mathcal{A} \text{ and } r \in \{1, 2, \cdots, |\mathcal{K}|\}. \tag{3.4}$$

Here, constraints (3.1) and (3.3) together ensure that for each $k \in \mathcal{K}$, the times that commodity k departs from its origin and arrives at its destination are both within the time window $[e^k, l^k]$, constraints (3.2) are due to the travel times of arcs on the flat path of each commodity k, and constraints (3.4) 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 (3.1)–(3.4).

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

$$H_n^k = \begin{cases} t_{\nu_n^k}^k - e^k, & \text{for } n = 1, \\ t_{\nu_n^k}^k - (t_{\nu_{n-1}^k}^k + \tau_{a_{n-1}^k}), & \text{for } n \in \{2, \ldots, m^k\}, \\ l^k - (t_{\nu_{n-1}^k}^k + \tau_{a_{n-1}^k}), & \text{for } n = m^k + 1. \end{cases}$$

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

$$\sum_{\alpha \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_\alpha \lceil \frac{\sum_{k \in C_r^\alpha} q^k}{u_\alpha} \rceil + \sum_{k \in \mathcal{K}} \sum_{n=1}^{m^k} c_{a_n^k}^k q^k + \sum_{k \in \mathcal{K}} \sum_{n=1}^{m^k+1} h_{\nu_n^k}^k q^k H_n^k,$$

where the first and the second terms are the total fixed cost and flow cost, respectively, 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

$$\begin{aligned} f(\mathcal{P}, \mathcal{C}) &= \sum_{\alpha \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_\alpha \lceil \frac{\sum_{k \in C_r^\alpha} q^k}{u_\alpha} \rceil + \sum_{k \in \mathcal{K}} \sum_{n=1}^{m^k} c_{a_n^k}^k q^k, \\ h(\mathcal{P}, \mathcal{T}) &= \sum_{k \in \mathcal{K}} \sum_{n=1}^{m^k+1} h_{\nu_n^k}^k q^k H_n^k. \end{aligned}$$

We assume that for each commodity $k \in \mathcal{K}$, the difference $(l^k - e^k)$ of 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-HC. Let $\mathbb{D}$ indicate the domain of all feasible solutions. The deterministic CTSNDP-HC can thus be formulated as follows:

$$\min_{(\mathcal{P}, \mathcal{C}, \mathcal{T}) \in \mathbb{D}} [f(\mathcal{P}, \mathcal{C}) + h(\mathcal{P}, \mathcal{T})]. \tag{3.5}$$

3.2.2 Robust CTSNDP-HC

We now introduce the robust CTSNDP-HC under travel time uncertainty, which we refer to as robust CTSNDP-HC for short. For this, we restrict the possible realized values of uncertain travel times on arcs by the following polyhedral uncertainty set.

Suppose that for every time when the service on an arc $\alpha \in \mathcal{A}$ is used to transport a shipment, the travel time $\tilde{\tau}_\alpha$ on α is uncertain and its realized value lies in the interval $[\overline{\tau}_\alpha - \hat{\tau}_\alpha, \overline{\tau}_\alpha + \hat{\tau}_\alpha]$, where $\overline{\tau}_\alpha \in \mathbb{N}_{>0}$ is the *nominal* value of $\tilde{\tau}_\alpha$, and $\hat{\tau}_\alpha \in \mathbb{N}_0$ with $\overline{\tau}_\alpha - \hat{\tau}_\alpha > 0$ denotes the maximum deviation of $\tilde{\tau}_\alpha$ with respect to the nominal value. As the result, $\tilde{\tau}_\alpha$ can be represented as:

$$\tilde{\tau}_\alpha = \overline{\tau}_\alpha + \hat{\tau}_\alpha \delta_\alpha, \text{ where } \delta_\alpha \in [-1, 1], \text{ for } \alpha \in \mathcal{A}.$$

For each $\alpha \in \mathcal{A}$, since there are at most $|\mathcal{K}|$ consolidations that can pass through arc α, we can use $\tilde{\tau}_{\alpha r}$ for $r \in \{1, 2, \ldots, |\mathcal{K}|\}$ to indicate the travel time of the r-th consolidation passing through arc α. Given an integer $\Gamma \in \mathbb{N}_0$ as the *budget of uncertainty*, which can be used to adjust the robustness level, we define the uncertainty set $\mathbb{U}(\Gamma)$ of travel times as follows:

$$\mathbb{U}(\Gamma) = \left\{ \tilde{\boldsymbol{\tau}} : \tilde{\tau}_{\alpha r} = \overline{\tau}_\alpha + \hat{\tau}_\alpha \delta_{\alpha r}, \sum_{\alpha \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} |\delta_{\alpha r}| \leq \Gamma, \delta_{\alpha r} \in [-1, 1], \forall \alpha \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\} \right\}. \tag{3.6}$$

The uncertainty set $\mathbb{U}(\Gamma)$ includes all the possible realized values for the vector of travel times, $\tilde{\boldsymbol{\tau}} = (\tilde{\tau}_{\alpha r})_{\alpha \in \mathcal{A}, r \in \{1,2,\ldots,|\mathcal{K}|\}}$, such that the total deviation of the travel times with respect to their nominal values does not exceed the given budget of uncertainty Γ.

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

$\mathcal{T}$ is a *wait-and-see* decision which is adaptive to the realized values of travel times. As a result, for different realized values of the travel times, the first-stage cost associated with $(\mathcal{P}, \mathcal{C})$ is deterministic, while both the second-stage decision $\mathcal{T}$ and its associated cost vary. The robust CTSNDP-HC aims to minimize the total cost of the two stages for the worst-case scenario over the uncertainty set of the travel times.

Specifically, let us first consider the second stage of the robust CTSNDP-HC. Given a flat solution $(\mathcal{P}, \mathcal{C})$ determined in the first stage, and after actual travel times $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$ are realized, one needs to determine a departure schedule $\mathcal{T} = (t^k_{\nu^k_n})_{k\in\mathcal{K},1\le n\le m^k}$, where each $t^k_{\nu^k_n}$ indicates the time that commodity k departs 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 $k \in C^{a^k_n}_{r(k,n)}$. 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 departure schedule $\mathcal{T}$ needs to satisfy constraints (3.1) due to the earliest available time e^k for $k \in \mathcal{K}$, constraints (3.4) due to the consolidations, and constraints (3.7) below:

$$t^k_{\nu^k_{n+1}} \ge t^k_{\nu^k_n} + \tilde{\tau}_{a^k_n,r(k,n)}, \text{ for } k \in \mathcal{K},\ n \in \{1, 2, \ldots, m^k - 1\}, \tag{3.7}$$

which are due to the actual travel times and similar to the constraints (3.2) for $k \in \mathcal{K}$ with $\tau_{a^k_n}$ replaced by $\tilde{\tau}_{a^k_n,r(k,n)}$. The domain of such departure schedules $\mathcal{T}$ is denoted by $\mathbb{T}(\mathcal{P}, \mathcal{C}, \tilde{\boldsymbol{\tau}})$.

Due to the travel time uncertainty, we relax the due time constraints in the second stage of the robust CTSNDP-HC. Instead, to restrict the violations of the due time constraints, we impose a penalty g^k per unit of time for the delay of commodity k's arrival at the destination d^k for each $k \in \mathcal{K}$. Let $g(\mathcal{P}, \mathcal{T})$ indicate the total delay penalty for a 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 arrival time of commodity k at the destination d^k. Hence, under the realized travel times $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$, the total cost, including the holding costs and delay penalties, is determined by $\mathcal{P}$ and $\mathcal{T}$, and it is equal to $h(\mathcal{P}, \mathcal{T}) + g(\mathcal{P}, \mathcal{T})$. Its minimum value, $\min_{\mathcal{T} \in \mathbb{T}(\mathcal{P}, \mathcal{C}, \tilde{\boldsymbol{\tau}})} [h(\mathcal{P}, \mathcal{T}) + g(\mathcal{P}, \mathcal{T})]\}$, is referred to as the second-stage cost of the robust CTSNDP-HC. Accordingly, the worst-case second-stage cost over all possible realized values of travel times in $\mathbb{U}(\Gamma)$ equals $\max_{\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)} \min_{\mathcal{T} \in \mathbb{T}(\mathcal{P}, \mathcal{C}, \tilde{\boldsymbol{\tau}})} [h(\mathcal{P}, \mathcal{T}) + g(\mathcal{P}, \mathcal{T})]$.

Next, consider the first-stage decisions of the robust CTSNDP-HC. Before the actual travel times are realized, one needs to determine a flat solution $(\mathcal{P}, \mathcal{C})$. As commonly required in practice, such a flat solution $(\mathcal{P}, \mathcal{C})$ needs to be timely-implementable for the nominal scenario where travel times take their nominal values. However, this cannot be guaranteed by the constraints imposed in the second stage of the robust CTSNDP-HC, where the due time constraints are relaxed. Therefore, we follow a *light robustness* approach, which was first proposed by Fischetti and Monaci [48] for robust optimization, to formulate the first-stage decisions of the CTSNDP-HC, requiring the first-stage decisions to be feasible for the deterministic CTSNDP-HC under the nominal scenario. Specifically, we require that the flat solution $(\mathcal{P}, \mathcal{C})$ to be determined in the first stage of the robust CTSNDP-HC must be timely-implementable under the nominal scenario, or in other words, it must satisfy that 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-HC under the nominal travel times. We refer to such a flat solution $(\mathcal{P}, \mathcal{C})$ as a nominal timely-implementable flat solution, and use $\mathbb{F}$ to indicate the domain of all nominal timely-implementable flat solutions. The deterministic first-stage total cost associated with $(\mathcal{P}, \mathcal{C})$ equals $f(\mathcal{P}, \mathcal{C})$. Accordingly, the robust CTSNDP-HC is to determine a nominal timely-implementable flat solution $(\mathcal{P}, \mathcal{C}) \in \mathbb{F}$ by minimizing the sum of the deterministic first-stage cost and the worst-case second-stage cost over the uncertainty set of travel times, as formulated below:

$$\min_{(\mathcal{P}, \mathcal{C}) \in \mathbb{F}} \{f(\mathcal{P}, \mathcal{C}) + \max_{\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)} \min_{\mathcal{T} \in \mathbb{T}(\mathcal{P}, \mathcal{C}, \tilde{\boldsymbol{\tau}})} [h(\mathcal{P}, \mathcal{T}) + g(\mathcal{P}, \mathcal{T})]\}. \tag{3.8}$$

3.3 Mixed Integer Programming Formulations

In this section, we start with a novel mixed integer linear programming (MILP) formulation that is based on flat network instead of time-expanded network for the deterministic CTSNDP-HC. From it, we derive a two-stage mixed integer nonlinear programming (MINLP) formulation for the robust CTSNDP-HC, which is linearized and solved in later sections.

3.3.1 MILP Formulation of Deterministic CTSNDP-HC

Instead of the TI model with variables and constraints indexed by time, here we introduce a new formulation utilizing a set of variables and constraints indexed by the consolidation indexes to model the temporal component of the deterministic CTSNDP-HC.

According to the problem description in Section 3.2, a feasible solution to the deterministic CTSNDP-HC 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 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 $r \in \{1, 2, \cdots, |\mathcal{K}|\}$ to represent the number of resource installations required by service on arc (i,j) to accommodate the commodities in consolidation $C_r^{(i,j)}$. To represent the departure schedule $\mathcal{T}$, we introduce a non-negative continuous variable v_{ij}^k for each $(i,j) \in \mathcal{A}$ and $k \in \mathcal{K}$ to represent the time when commodity k departs from node i when passing through arc (i,j), which equals 0 if commodity k does not pass through arc (i,j). We also introduce a non-negative continuous variable b_{ijr} for each $(i,j) \in \mathcal{A}$ and $r \in \{|\mathcal{K}|\}$ to represent the time when the shipment of the r-th consolidation $C_r^{(i,j)}$ on arc (i,j) departs from node i. Moreover, we use 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, which equals 0 if commodity

k does not pass node i. Accordingly, we can formulate the deterministic CTSNDP-HC by the following MILP model, which is referred to model DO, where M_1 denotes a sufficiently large constant:

$$[\text{DO}] \ \min \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k)\cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|} f_{ij}\cdot y_{ijr} + \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}} (h_i^k q^k)\cdot w_i^k \tag{3.9}$$

$$\text{s.t.} \sum_{(i,j)\in\mathcal{A}} x_{ij}^k - \sum_{(j,i)\in\mathcal{A}} x_{ji}^k = \begin{cases} 1 & i = o^k, \\ -1 & i = d^k, \\ 0 & \text{otherwise} \end{cases} \quad \forall k\in\mathcal{K}, i\in\mathcal{N}, \tag{3.10}$$

$$\sum_{k\in\mathcal{K}} q^k z_{ijr}^k \le u_{ij} y_{ijr}, \qquad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.11}$$

$$\sum_{r=1}^{|\mathcal{K}|} z_{ijr}^k = x_{ij}^k, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.12}$$

$$\sum_{j:(j,i)\in\mathcal{A}} (v_{ji}^k + \tau_{ji} x_{ji}^k) \le \sum_{j:(i,j)\in\mathcal{A}} v_{ij}^k, \qquad \forall\ i\in\mathcal{N}\setminus\{o^k, d^k\}, k\in\mathcal{K}, \tag{3.13}$$

$$\sum_{j:(o^k,j)\in\mathcal{A}} v_{o^k j}^k \ge e^k, \qquad \forall\ k\in\mathcal{K}, \tag{3.14}$$

$$\sum_{j:(j,d^k)\in\mathcal{A}} (v_{jd^k}^k + \tau_{jd^k} x_{jd^k}^k) \le l^k, \qquad \forall\ k\in\mathcal{K}, \tag{3.15}$$

$$v_{ij}^k \le M_1 x_{ij}^k, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.16}$$

$$v_{ij}^k \le b_{ijr} + M_1(1 - z_{ijr}^k), \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.17}$$

$$v_{ij}^k \ge b_{ijr} - M_1(1 - z_{ijr}^k), \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.18}$$

$$w_i^k = \begin{cases} \sum_{j:(i,j)\in\mathcal{A}} v_{ij}^k - e^k & i = o^k, \\ l^k - \sum_{j:(j,i)\in\mathcal{A}} (v_{ji}^k + \tau_{ji} x_{ji}^k) & i = d^k, \\ \sum_{j:(i,j)\in\mathcal{A}} v_{ij}^k - \sum_{j:(j,i)\in\mathcal{A}} (v_{ji}^k + \tau_{ji} x_{ji}^k) & \text{otherwise}, \end{cases} \quad \forall\ i\in\mathcal{N}, \forall\ k\in\mathcal{K}, \tag{3.19}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.20}$$

$$y_{ijr} \in \mathbb{N}_{\ge 0}, \qquad \forall\ (i,j)\in\mathcal{A}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.21}$$

$$z_{ijr}^k \in \{0,1\}, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, r\in\{1,2,...,|\mathcal{K}|\}, \tag{3.22}$$

$$v_{ij}^k \ge 0, \qquad \forall\ (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{3.23}$$

$$b_{ijr} \geq 0, \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.24}$$

$$w_i^k \geq 0, \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}. \tag{3.25}$$

In model DO, the objective function (3.9) indicates the total cost to be minimized, which includes three terms for the total transportation cost, total fixed cost, and total holding cost, respectively. Constraints (3.10)–(3.12) are imposed to define the routing and the consolidation plans. Specifically, constraints (3.10) are *flow balance constraints*, ensuring that each commodity travels along one elementary flat path from its origin to its destination. Constraints (3.11) are *capacity constraints*, ensuring that the total quantity of commodities in each consolidation of an arc does not exceed the total service capacity installed on the arc. Constraints (3.12) are *consolidation coverage constraints*, ensuring that for every arc (i, j) on the flat path of commodity k, where $k \in \mathcal{K}$, there must be a consolidation of arc (i, j) that contains k. Constraints (3.13)–(3.19) are imposed to define the departure schedule. Specifically, constraints (3.13)–(3.15) are imposed on commodities' departure times with respect to the travel time of each arc, the earliest available time of each commodity, and the due time of each commodity. Constraints (3.16) ensure that for each commodity, its departure time for every unvisited node is zero. Constraints (3.17) and (3.18) ensure that for each arc (i, j), the commodities that are consolidated to be shipped together through (i, j) has the same departure time for node i. Constraints (3.19) are imposed to define the holding time for each node i and commodity k based on the departure schedule and the routing plan.

It can be seen that 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 $(\mathcal{P}, \mathcal{C})$, $\boldsymbol{v}$ corresponds to a departure schedule $\mathcal{T}$ that satisfies (3.1)–(3.4), and thus, such $(\mathcal{P}, \mathcal{C}, \mathcal{T})$ forms a feasible solution to the deterministic CTSNDP-HC. Note that model DO is equivalent to the complete TI model proved in Section 2.2. Model DO is the first MILP formulation of the deterministic CTSNDP-HC that models the temporal component of the problem based on the consolidation indexes, which is referred to as the *consolidation-index* formulation.

3.3.2 Two-Stage MINLP Formulation of Robust CTSNDP-HC

We can extend model DO of the deterministic CTSNDP-HC to formulate the robust CTSNDP-HC by a two-stage MINLP model, where the first stage determines the routing and the consolidation plans indicated by $(\boldsymbol{x}, \boldsymbol{z})$ before the actual travel times are realized, and the second stage determines the departure schedule and holding times indicated by $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w})$ after the actual travel times are realized.

As required in the first stage of the robust CTSNDP-HC, $(\boldsymbol{x}, \boldsymbol{z})$ needs to ensure the existence of a departure schedule that satisfies the constraints with respect to the commodities' earliest available times and due times under the nominal scenario. Thus, 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, which are similar to variables v_{ij}^k and b_{ijr} of model DO. Moreover, as required in the second stage of the robust CTSNDP-HC, constraints with respect to the commodities' due times are relaxed, but delay penalties are imposed. Thus, we need to introduce a new decision variable s^k for each $k \in \mathcal{K}$, indicating the delay of the commodity k's arrival at its destination.

With the above decision variables introduced, the two-stage MINLP model for the robust CTSNDP-HC, which is referred to as model RO, is presented below:

$$[\text{RO}] \quad \min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + F_{RP}(\boldsymbol{x}, \boldsymbol{z}) \tag{3.26}$$

$$\text{s.t. } (3.10) - (3.12), (3.20) - (3.22) \tag{3.27}$$

$$\sum_{j:(j,i) \in \mathcal{A}} (\overline{v}_{ji}^k + \overline{\tau}_{ji} x_{ji}^k) \leq \sum_{j:(i,j) \in \mathcal{A}} \overline{v}_{ij}^k, \qquad \forall\, i \in \mathcal{N} \setminus \{o^k, d^k\}, k \in \mathcal{K}, \tag{3.28}$$

$$\sum_{j:(o^k,j) \in \mathcal{A}} \overline{v}_{o^k j}^k \geq e^k, \qquad \forall\, k \in \mathcal{K}, \tag{3.29}$$

$$\sum_{j:(j,d^k) \in \mathcal{A}} (\overline{v}_{jd^k}^k + \overline{\tau}_{jd^k} x_{id^k}^k) \leq l^k, \qquad \forall\, k \in \mathcal{K}, \tag{3.30}$$

$$\overline{v}_{ij}^k \leq M x_{ij}^k, \qquad \forall\, (i,j) \in \mathcal{A}, k \in \mathcal{K}, \tag{3.31}$$

$$\overline{v}_{ij}^k \leq \overline{b}_{ijr} + M(1 - z_{ijr}^k), \qquad \forall\, (i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.32}$$

$$\overline{v}_{ij}^k \geq \overline{b}_{ijr} - M(1 - z_{ijr}^k), \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.33}$$

$$\overline{v}_{ij}^k \geq 0, \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, \tag{3.34}$$

$$\overline{b}_{ijr} \geq 0, \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}. \tag{3.35}$$

where $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ indicates the worst-case second-stage cost and is defined by the following max-min optimization model:

$$[\mathrm{SO}(\boldsymbol{x}, \boldsymbol{z})] \quad F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \max_{\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)} \ \min \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} (h_i^k q^k) \cdot w_i^k + \sum_{k \in \mathcal{K}} g^k \cdot s^k \tag{3.36}$$

$$\text{s.t.} \sum_{j:(j,i) \in \mathcal{A}} (v_{ji}^k + \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z_{jir}^k) \leq \sum_{j:(i,j) \in \mathcal{A}} v_{ij}^k, \qquad \forall\ i \in \mathcal{N} \setminus \{o^k, d^k\}, k \in \mathcal{K}, \tag{3.37}$$

$$\sum_{j:(o^k,j) \in \mathcal{A}} v_{o^k j}^k \geq e^k, \qquad \forall\ k \in \mathcal{K}, \tag{3.38}$$

$$\sum_{j:(j,d^k) \in \mathcal{A}} (v_{jd^k}^k + \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jd^k r} z_{jd^k r}^k) \leq l^k + s^k, \qquad \forall\ k \in \mathcal{K}, \tag{3.39}$$

$$v_{ij}^k \leq M_1 x_{ij}^k, \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, \tag{3.40}$$

$$v_{ij}^k \leq b_{ijr} + M_1(1 - z_{ijr}^k), \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.41}$$

$$v_{ij}^k \geq b_{ijr} - M_1(1 - z_{ijr}^k), \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.42}$$

$$w_i^k \geq \begin{cases} \sum_{j:(i,j) \in \mathcal{A}} v_{ij}^k - e^k, & i = o^k, \\ (l^k + s^k) - \sum_{j:(j,i) \in \mathcal{A}} (v_{ji}^k + \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z_{jir}^k), & i = d^k, \qquad \forall\ i \in \mathcal{N}, \forall\ k \in \mathcal{K}, \\ \sum_{j:(i,j) \in \mathcal{A}} v_{ij}^k - \sum_{j:(j,i) \in \mathcal{A}} (v_{ji}^k + \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z_{jir}^k), & \text{otherwise}, \end{cases} \tag{3.43}$$

$$v_{ij}^k \geq 0, \qquad \forall\ (i,j) \in \mathcal{A}, k \in \mathcal{K}, \tag{3.44}$$

$$b_{ijr} \geq 0, \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.45}$$

$$w_i^k \geq 0, \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}, \tag{3.46}$$

$$s^k \geq 0, \qquad \forall k \in \mathcal{K}. \tag{3.47}$$

The objective (3.26) of model RO is to minimize the sum of the deterministic first-stage cost, including the transportation costs and the fixed installation costs shown in the first two terms of (3.26), and the worst-case second-stage cost over the uncertainty set $\mathbb{U}(\Gamma)$ of travel times, represented by $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$. In model RO, constraints in (3.27) are the same as those of model DO imposed on $(\boldsymbol{x}, \boldsymbol{z})$. Constraints (3.28)–(3.35) are similar to (3.13)–(3.18), (3.23), and (3.24) 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.

Given any $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$, the inner minimization model of the max-min model (3.36)–(3.47) for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ formulates the second stage of the robust CTSNDP-HC. It formulates the second-stage cost by a linear programming (LP) model that aims to determine $(\mathbf{v}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{s})$ with the sum of the holding costs and delay penalties minimized. Most of its constraints are the same as those of model DO imposed on $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w})$, except (3.37), (3.39) and (3.43). Compared with constraints (3.13), (3.15), and (3.19) of model DO, constraints (3.37), (3.39), and (3.43) replace $\tau_{ji} x_{ji}^k$ with $\sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z_{jir}^k$ for $(j, i) \in \mathcal{A}$, 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 (3.39) and (3.43) so that it equals the delay of commodity k's arrival at its destination.

3.4 A Column-and-Constraint Generation Solution Method

The column-and-constraint generation (C&CG) method proposed by Zeng and Zhao [111] is widely used in solving two-stage robust optimization models, including the two-stage robust optimization model for service network design under demand uncertainty [108]. It has also been shown to be more effectively than the standard Benders decomposition approach, which iteratively includes new Benders cuts. Thus, we consider to solve the two-stage robust CTSNDP-HC under travel time uncertainty with a C&CG method.

Denote $\mathcal{X}$ as the domain of variables $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}})$ defined by linear constraints (3.27)–(3.35) and $\mathcal{Q}(\tilde{\boldsymbol{\tau}})$ as the domain of variables $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{s})$ defined by linear constraints (3.37)–(3.47) under travel time realization $\tilde{\boldsymbol{\tau}}$. Model RO proposed in Section 3.3.2 can be rewritten as the following MILP:

$$
\begin{aligned}
[\text{ROLP}]\ \min \quad & \sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k)\cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}}\sum_{r=1}^{|\mathcal{K}|} f_{ij}\cdot y_{ijr} + \phi && (3.48)\\
s.t. \quad & \phi \geq \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}} (h_i^k q^k)\cdot w_i^{k(\tilde{\boldsymbol{\tau}})} + \sum_{k\in\mathcal{K}} g^k\cdot s^{k(\tilde{\boldsymbol{\tau}})}, \quad \forall\, \tilde{\boldsymbol{\tau}}\in\mathbb{U}(\Gamma), && (3.49)\\
& (\boldsymbol{v}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{b}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{w}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{s}^{(\tilde{\boldsymbol{\tau}})}) \in \mathcal{Q}(\tilde{\boldsymbol{\tau}}), \quad \forall\, \tilde{\boldsymbol{\tau}}\in\mathbb{U}(\Gamma), && (3.50)\\
& (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}}) \in \mathcal{X}. && (3.51)
\end{aligned}
$$

where ϕ is epigraphical variable introduced.

As a result, solving the two-stage robust CTSNDP-HC reduces to optimizing a mixed integer programming. It is straightforward that a partial formulation of model ROLP defined over a subset of $\mathbb{U}(\Gamma)$ is a valid relaxation of the original model ROLP. Stronger lower bounds can be obtained by gradually adding scenarios into the subset of $\mathbb{U}(\Gamma)$ to expand the partial formulation. The C&CG method can obtain the optimal solution by iteratively adding new scenarios $\tilde{\boldsymbol{\tau}}$ as well as new cuts in (3.49) and (3.50). We then show how we identify the new worst-case scenarios that should be added into the scenario subset in Section 3.4.1 and illustrate the whole framework of the C&CG method in Section 3.4.2.

3.4.1 MILP Reformulation of the Second-Stage Problem

For any given $(\boldsymbol{x}, \boldsymbol{z})$, its worst-case second-stage cost $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ and the corresponding worst-case scenario can be obtained by solving model SO$(\boldsymbol{x}, \boldsymbol{z})$ defined by (3.36)–(3.47). The obtained worst-case scenario can be identified and added into the subset of $\mathbb{U}(\Gamma)$ as a new scenario. However, model SO$(\boldsymbol{x}, \boldsymbol{z})$ is a max-min optimization model hard to solve directly. We show as follows that model $\text{SO}(\boldsymbol{x}, \boldsymbol{z})$ can be reformulated as a maximization MILP model, which is much more tractable.

First, consider the inner minimization problem of model $\text{SO}(\boldsymbol{x}, \boldsymbol{z})$ under any given $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$, which formulates the second stage of the robust CTSNDP-HC. As shown below, it is a linear programming, and we refer to it as model $\text{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$:

$$\begin{aligned} [\text{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})] \ \min \quad & \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h_i^k q^k) \cdot w_i^k + \sum_{k\in\mathcal{K}} g^k \cdot s^k \\ \text{s.t.} \quad & (3.37) - (3.47). \end{aligned}$$

Lemma 1 below shows that for each $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$, model $\text{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$ always has a feasible solution if $(\boldsymbol{x}, \boldsymbol{z})$ satisfies constraints (3.27)–(3.35) of model RO.

Lemma 1. *For any $(\boldsymbol{x}, \boldsymbol{z})$ that satisfies constraints (3.27)–(3.35) of model RO, and for any $\hat{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$, model $\text{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$ always has a feasible solution.*

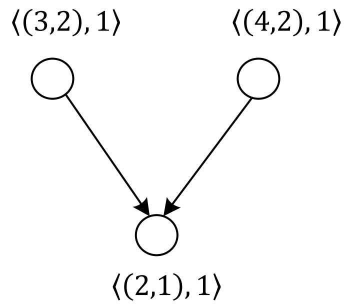
Proof. For any given $(\boldsymbol{x}, \boldsymbol{z})$ that satisfies constraints (3.27)–(3.35) of model RO, it corresponds to a nominal timely-implementable flat solution $(\mathcal{P}, \mathcal{C})$. Consider any realized travel time $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$. For such $(\mathcal{P}, \mathcal{C})$ and $\tilde{\boldsymbol{\tau}}$, we first show as follows that there exists a departure schedule $\mathcal{T}$ such that constraints (3.1)–(3.4) are satisfied, from which we can then obtain a feasible solution to model $\text{LP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$.

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 $(a_1^k, ..., a_{m^k}^k)$. For each $n \in \{1, 2, \ldots, m^k\}$, there must exist a consolidation $C_{r_n^k}^{a_n^k} \in \mathcal{C}$ for arc a_n^k with $r_n^k \in \{1, 2, \cdots, |\mathcal{K}|\}$ such that $k \in C_{r_n^k}^{a_n^k}$. We can now construct a network $\mathcal{G}_{\mathcal{C}} = \{\mathcal{N}_{\mathcal{C}}, \mathcal{A}_{\mathcal{C}}\}$ where each non-empty consolidation $C_r^\alpha \in \mathcal{C}$ corresponds to a node, denoted by $\langle \alpha, r \rangle$, in the node set $\mathcal{N}_{\mathcal{C}}$, and each pair of consolidations $C_{r_n^k}^{a_n^k}$ and $C_{r_{n+1}^k}^{a_{n+1}^k}$ for $k \in \mathcal{K}$ and $n \in \{1, ..., m^k - 1\}$ corresponds to an arc $(\langle a_n^k, r_n^k \rangle, \langle a_{n+1}^k, r_{n+1}^k \rangle)$ in the arc set $\mathcal{A}_{\mathcal{C}}$. See Figure 3.1 for an example of such a network $\mathcal{G}_{\mathcal{C}}$.

Since the flat solution $(\mathcal{P}, \mathcal{C})$ is nominal timely-implementable, there exists a departure schedule $\mathcal{T}$ which satisfies (3.1)–(3.4) 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

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

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



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

Figure 3.1 An Example of network $\mathcal{G}_\mathcal{C}$ constructed from a given nominal timely-implementable flat solution $(\mathcal{P}, \mathcal{C})$

corresponding departure time from node ν, which is denoted by $t_{\alpha,r}$. For each pair of consolidations $C^{a_n^k}_{r_n^k}$ and $C^{a_{n+1}^k}_{r_{n+1}^k}$ with $k \in \mathcal{K}$ and $n \in \{1, ..., m^k - 1\}$, the departure time of $C^{a_n^k}_{r_n^k}$ 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 $C^{a_{n+1}^k}_{r_{n+1}^k}$ 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

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

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 $\tilde{\boldsymbol{\tau}} \in \mathbb{U}(\Gamma)$. For $n = 1, 2, \ldots, |\mathcal{N}_\mathcal{G}|$, we can now set the departure time of consolidation $C^{\alpha_n}_{r_n}$, which is denoted by $\hat{t}_{\alpha_n, r_n}$, iteratively as follows:

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

It can be seen that for each commodity $k \in \mathcal{K}$,

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

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 (3.1), (3.2) and (3.4) under the travel time $\tilde{\boldsymbol{\tau}}$. 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, by their definitions, which form a feasible solution to model LP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$. Hence, Lemma 1 is proved. $\square$

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 (3.37)–(3.43), respectively. By Lemma 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 programming below, which is referred to as model DLP$(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})$:

$$\begin{aligned}
[\text{DLP}(\boldsymbol{x}, \boldsymbol{z}, \tilde{\boldsymbol{\tau}})] \max_{(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\lambda})} & \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N} \setminus \{o^k, d^k\}} \Big(\sum_{j:(j,i) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z_{jir}^k \Big) \cdot \beta_i^k + \sum_{k \in \mathcal{K}} e^k \cdot \gamma^k \\
& + \sum_{k \in \mathcal{K}} \Big(\sum_{(j,d^k) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jd^k r} z_{jd^k r}^k - l^k \Big) \cdot \psi^k - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k \\
& - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1 (1 - z_{ijr}^k)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k) - \sum_{k \in \mathcal{K}} e^k \cdot \lambda_{o^k}^k \\
& + \sum_{k \in \mathcal{K}} \Big(l^k - \sum_{j:(j,d^k) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jd^k r} z_{jd^k r}^k \Big) \cdot \lambda_{d^k}^k - \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N} \setminus \{o^k, d^k\}} \Big(\sum_{j:(j,i) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \tilde{\tau}_{jir} z_{jir}^k \Big) \cdot \lambda_i^k
\end{aligned} \tag{3.52}$$

$$\text{s.t.} \quad \beta_i^k - \beta_j^k - \eta_{ij}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{ijr}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{ijr}^k - \lambda_i^k + \lambda_j^k \leq 0, \ \forall\, k \in \mathcal{K}, (i,j) \in \mathcal{A}, i \neq o^k, j \neq d^k, \tag{3.53}$$

$$-\beta_j^k + \gamma^k - \eta_{o^k j}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{o^k jr}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{o^k jr}^k - \lambda_{o^k}^k + \lambda_j^k \leq 0, \ \forall\, k \in \mathcal{K}, (o^k, j) \in \mathcal{A}, j \neq d^k, \tag{3.54}$$

$$\beta_i^k - \psi^k - \eta_{id^k}^k - \sum_{r=1}^{|\mathcal{K}|} \theta_{id^k r}^k + \sum_{r=1}^{|\mathcal{K}|} \xi_{id^k r}^k - \lambda_i^k + \lambda_{d^k}^k \leq 0, \ \forall\, k \in \mathcal{K}, (i, d^k) \in \mathcal{A}, i \neq o^k, \tag{3.55}$$

$$\gamma^k - \psi^k - \eta^k_{o^k d^k} - \sum_{r=1}^{|\mathcal{K}|} \theta^k_{o^k d^k r} + \sum_{r=1}^{|\mathcal{K}|} \xi^k_{o^k d^k r} - \lambda^k_{o^k} + \lambda^k_{d^k} \leq 0,\ \forall\ k \in \mathcal{K}, (o^k, d^k) \in \mathcal{A}, \tag{3.56}$$

$$\sum_{k\in\mathcal{K}} \theta^k_{ijr} - \sum_{k\in\mathcal{K}} \xi^k_{ijr} \leq 0,\ \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.57}$$

$$\lambda^k_i \leq h^k_i q^k,\ \forall\ i \in \mathcal{N}, k \in \mathcal{K}, \tag{3.58}$$

$$\psi^k - \lambda^k_{d^k} \leq g^k,\ \forall\ k \in \mathcal{K}, \tag{3.59}$$

$$\boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}, \boldsymbol{\psi} \geq \mathbf{0}, \boldsymbol{\eta} \geq \mathbf{0}, \boldsymbol{\theta} \geq \mathbf{0}, \boldsymbol{\xi} \geq \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}. \tag{3.60}$$

This, together with the definition of uncertainty set $\mathbb{U}(\Gamma)$ in (3.6), implies that $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ can be reformulated as the following maximization MINLP model:

$$\begin{aligned}
F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \quad & \max_{(\tilde{\boldsymbol{\tau}}, \boldsymbol{\delta}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\lambda})} \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i) \Big) \cdot \tilde{\tau}_{jir} \\
& - \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (M_1 x^k_{ij}) \cdot \eta^k_{ij} + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1(z^k_{ijr} - 1)] \cdot (\theta^k_{ijr} + \xi^k_{ijr}) \\
& + \sum_{k\in\mathcal{K}} e^k \cdot (\gamma^k - \lambda^k_{o^k}) + \sum_{k\in\mathcal{K}} l^k \cdot (\lambda^k_{d^k} - \psi^k) && (3.61) \\
\text{s.t.} \quad & (3.53) - (3.60) && (3.62) \\
& \tilde{\tau}_{ijr} = \overline{\tau}_{ij} + \hat{\tau}_{ij}\delta_{ijr}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, && (3.63) \\
& -1 \leq \delta_{ijr} \leq 1, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, && (3.64) \\
& \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} |\delta_{ijr}| \leq \Gamma. && (3.65)
\end{aligned}$$

where $\mathcal{K}_i = \{k \in \mathcal{K} : i \neq o^k \text{ and } i \neq d^k\}$, $\mathcal{K}^d_i = \{k \in \mathcal{K} : i = d^k\}$. The above maximization MINLP model for $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ can further be linearized and reformulated as a MILP model based on Proposition 3 below.

Proposition 3. *There exists an optimal solution to the maximization MINLP model defined in (3.61)–(3.65) such that $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$.*

Proof. For any given $(\boldsymbol{x}, \boldsymbol{z})$ and fixed $(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\lambda})$, with $\tilde{\tau}_{ijr}$ replaced by $\overline{\tau}_{ij} + \hat{\tau}_{ij}\delta_{ijr}$ for $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, \cdots, |\mathcal{K}|\}$, model (3.61)–(3.65) becomes an LP model on $\boldsymbol{\delta}$ subject to constraints (3.64) and (3.65). It is well known that for any LP model, there exists an optimal solution that is an extreme point of its feasible region. For the feasible region defined by constraints (3.64) and (3.65), we can see that its extreme point must satisfy that $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. This implies that there exists an optimal solution to model (3.61)–(3.65) that satisfies $\delta_{ijr} \in \{-1, 0, 1\}$ for each $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. Hence, Proposition 3 is proved. $\square$

By Proposition 3, constraints (3.64) can be replaced with $\delta_{ijr} \in \{-1, 0, 1\}$ for $(i, j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. Accordingly, (3.63) implies 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 can replace each nonlinear term $\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big) \cdot \tilde{\tau}_{jir}$ with a new variable φ_{jir} and 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, for all $(j, i) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. Let $\tilde{\tau}_{ijr,-1} = \overline{\tau}_{ijr} - \hat{\tau}_{ijr}$, $\tilde{\tau}_{ijr,0} = \overline{\tau}_{ijr}$ and $\tilde{\tau}_{ijr,1} = \overline{\tau}_{ijr} + \hat{\tau}_{ijr}$. We introduce the following constraints to ensure that $\varphi_{jir} = \Big(\sum\limits_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum\limits_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big) \cdot \tilde{\tau}_{jir}$ for all $(j, i) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$.

$$\zeta_{ijr,-1} + \zeta_{ijr,0} + \zeta_{ijr,1} = 1, \quad \forall\ (i, j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.66}$$

$$\begin{aligned}
&\Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big)\tilde{\tau}_{jir,\ell} - M_2(1 - \zeta_{jir,\ell}) \leq \varphi_{jir} \\
&\qquad \leq \Big(\sum_{k\in\mathcal{K}_i} z^k_{jir}(\beta^k_i - \lambda^k_i) + \sum_{k\in\mathcal{K}^d_i} z^k_{jir}(\psi^k - \lambda^k_i)\Big)\tilde{\tau}_{jir,\ell} + M_2(1 - \zeta_{jir,\ell}), \\
&\qquad\qquad \forall\ (j, i) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \ell \in \{-1, 0, 1\},
\end{aligned} \tag{3.67}$$

$$\zeta_{ijr,-1} \in \{0, 1\}, \zeta_{ijr,0} \in \{0, 1\}, \zeta_{ijr,1} \in \{0, 1\}, \quad \forall\ (i, j) \in \mathcal{A}, r \in \{1, 2, ..., |\mathcal{K}|\}, \tag{3.68}$$

where M_2 is a sufficiently large constant. Accordingly, constraints (3.65) can be replaced with a linear constraint shown in (3.69) below:

$$\sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} (\zeta_{ijr,-1} + \zeta_{ijr,1}) \leq \Gamma. \tag{3.69}$$

Accordingly, the maximization MINLP model SO($\boldsymbol{x}, \boldsymbol{z}$) defined by (3.61)–(3.65) can be further reformulated as the following maximization MILP model:

$$
\begin{aligned}
[\mathrm{SRP}(\boldsymbol{x}, \boldsymbol{z})]\ F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \quad & \max_{(\boldsymbol{\zeta}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\varphi})} \sum_{(j,i)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} \varphi_{jir} - \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (M_1 x_{ij}^k) \cdot \eta_{ij}^k \\
& + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} [M_1(z_{ijr}^k - 1)] \cdot (\theta_{ijr}^k + \xi_{ijr}^k) \\
& + \sum_{k\in\mathcal{K}} e^k \cdot (\gamma^k - \lambda_{o^k}^k) + \sum_{k\in\mathcal{K}} l^k \cdot (\lambda_{d^k}^k - \psi^k) \qquad (3.70) \\
\text{s.t.} \quad & (3.53) - (3.60),\ (3.66) - (3.68) \text{ and } (3.69). \qquad (3.71)
\end{aligned}
$$

The above reformulation can be solved directly by the solver to obtain the worst-case second-stage cost $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ and the corresponding worst-case scenario $\tilde{\boldsymbol{\tau}}$ can be obtained by below (3.72).

$$
\tilde{\tau}_{ijr} = \tilde{\tau}_{ijr,-1}\zeta_{ijr,-1} + \tilde{\tau}_{ijr,0}\zeta_{ijr,0} + \tilde{\tau}_{ijr,1}\zeta_{ijr,1}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}, \quad (3.72)
$$

which, by Proposition 3, implies that

$$
\tilde{\tau}_{ijr} \in \{\overline{\tau}_{ijr} - \hat{\tau}_{ijr}, \overline{\tau}_{ijr}, \overline{\tau}_{ijr} + \hat{\tau}_{ijr}\}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}. \qquad (3.73)
$$

3.4.2 Algorithm Framework

From model ROLP over a scenario subset $\Lambda \subseteq \mathbb{U}(\Gamma)$, we can define the master problem of the C&CG method as below, which is referred to as model $\mathrm{MP}_{C\&CG}$:

$$
\begin{aligned}
[\mathrm{MP}_{C\&CG}]\ \min \quad & \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + \phi \qquad (3.74) \\
s.t. \quad & \phi \geq \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h_i^k q^k) \cdot w_i^{k(\tilde{\boldsymbol{\tau}})} + \sum_{k\in\mathcal{K}} g^k \cdot s^{k(\tilde{\boldsymbol{\tau}})}, \quad \forall\ \tilde{\boldsymbol{\tau}} \in \Lambda, \quad (3.75) \\
& (\boldsymbol{v}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{b}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{w}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{s}^{(\tilde{\boldsymbol{\tau}})}) \in \mathcal{Q}(\tilde{\boldsymbol{\tau}}), \quad \forall\ \tilde{\boldsymbol{\tau}} \in \Lambda, \qquad (3.76) \\
& (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}}) \in \mathcal{X}. \qquad (3.77)
\end{aligned}
$$

The C&CG method iteratively solves the master problem to obtain a first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, solves the corresponding subproblem SRP$(\boldsymbol{x}, \boldsymbol{z})$ to identify a new worst-case scenario $\tilde{\boldsymbol{\tau}}$, and then adds $\tilde{\boldsymbol{\tau}}$ to expand the scenario subset Λ so as to update the master problem MP$_{C\&CG}$. With a new scenario $\tilde{\boldsymbol{\tau}}$ added to Λ, new decision variables $(\boldsymbol{v}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{b}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{w}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{s}^{(\tilde{\boldsymbol{\tau}})})$, together with the corresponding constraints in (3.75)–(3.76) defined on them, are added to the master problem MP$_{C\&CG}$.

Define the upper and lower bounds of the optimal objective value to the original problem as UB and LB, respectively. The framework of our C&CG method for the robust CTSNDP-HC can be summarized as below:

Step 1: Set $n = 0$, $UB = +\infty$ and $LB = -\infty$.

Step 2: Solve the master problem MP$_{C\&CG}$ to find its optimal solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \phi)$, and let LB equal to the optimal objective value of the MP$_{C\&CG}$.

Step 3: Solve the subproblem SRP$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ to obtain its optimal objective value $F_{RP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ as well as the worst-case scenario $\tilde{\boldsymbol{\tau}}^{(n)}$ according to (3.72), and update

$$UB = \min\{UB, \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot \hat{x}_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot \hat{y}_{ijr} + F_{RP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})\}$$

Step 4: If a certain stopping condition is reached, then the algorithm stops. Otherwise, update $\Lambda = \Lambda \bigcup \{\tilde{\boldsymbol{\tau}}^{(n)}\}$. Add new decision variables $(\boldsymbol{v}^{(\tilde{\boldsymbol{\tau}}^{(n)})}, \boldsymbol{b}^{(\tilde{\boldsymbol{\tau}}^{(n)})}, \boldsymbol{w}^{(\tilde{\boldsymbol{\tau}}^{(n)})}, \boldsymbol{s}^{(\tilde{\boldsymbol{\tau}}^{(n)})})$ as well as their corresponding constraints in (3.75) – (3.76) to the master problem MP$_{C\&CG}$. Update $n = n + 1$ and then go to Step 2.

At each iteration of the C&CG method, the UB and LB are updated by solving the corresponding master problem and subproblem, while a new worst-case scenario $\tilde{\boldsymbol{\tau}}$ in $\mathbb{U}(\Gamma)$ is also found and added into the scenario subset Λ. The C&CG method stops when a certain condition is satisfied. A common stopping condition is that the optimality gap $(UB - LB)/UB = 0$. Under this stopping condition, we can see as follows that our C&CG method must converge to an optimal solution to the robust CTSNDP-HC in a finite number of iterations. To see this, it is worthy to note that if the worst-case scenario $\tilde{\boldsymbol{\tau}}^{(n)}$ identified in Step 3 is in the current scenario subset Λ, by an argument similar to that in Zeng and Zhao [111] we know that $LB = UB$, implying that the stopping condition $(UB - LB)/UB = 0$ is satisfied. Moreover, for each of such scenarios $\tilde{\boldsymbol{\tau}}$ identified in Step 3, it satisfies that $\tilde{\tau}_{ijr} \in \{\overline{\tau}_{ijr} - \hat{\tau}_{ijr}, \overline{\tau}_{ijr}, \overline{\tau}_{ijr} + \hat{\tau}_{ijr}\}$, $\forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}$ by (3.73). Hence, since the number of such scenarios $\tilde{\boldsymbol{\tau}}$ is finite, we obtain that our C&CG method must reach $(UB - LB)/UB = 0$ in a finite number of iterations. This implies that under the stopping condition $(UB - LB)/UB = 0$, our C&CG method must converge to an optimal solution to the robust CTSNDP-HC in a finite number of iterations.

We also note that this C&CG method can be extended to solve the cases with non-integer budget of uncertainty Γ. For the robust CTSNDP with non-integer Γ, we have a new proposition by extending the result of Proposition 3, that is, there exists an optimal solution to the maximization MINLP model defined in (3.61)–(3.65) such that $\delta_{ijr} \in \{-1, 0, 1, \Gamma - \lfloor \Gamma \rfloor, -(\Gamma - \lfloor \Gamma \rfloor)\}$ for each $(i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$. Based on this new proposition, the linearization method for model SO and the C&CG solution method are still valid for the robust CTSNDP with non-integer Γ. Moreover, our robust optimization formulation RO and the C&CG solution method can be directly applied to the cases with strict robustness, i.e., the robust solution should be feasible for all possible scenarios of the given uncertainty set, by using an infinity large delay penalty $\boldsymbol{g}$.

3.5 An Enhanced Column-and-Constraint Generation Solution Method

The performance of the C&CG method relies on the efficiency of solving the master problem and the subproblem in each iteration. In the C&CG method presented in Section 3.4, the model sizes of the master problems and the subproblems, and hence the running time spent in solving them, grow with the value of $|\mathcal{K}|$, which indicates the maximum total number of consolidations allowed on each arc. However, the actual total number of consolidations on each arc $(i,j) \in \mathcal{A}$ used in the optimal solution is often expected to be much smaller than $|\mathcal{K}|$, particularly for those large-scale instances with relatively large $|\mathcal{K}|$.

Consider a flat solution $(\mathcal{P}, \mathcal{C})$ and the sets $\mathcal{C}^\alpha$, $\forall \alpha \in \mathcal{A}$, where each $\mathcal{C}^\alpha$ consists of the non-empty consolidations on arc α derived from $\mathcal{C}$. It is clear that excluding empty consolidation sets from $\mathcal{C}$ does not change the execution of the routing and consolidation plans, and thus it does change the corresponding worst-case second-stage cost. This implies that the flat solution $(\mathcal{P}, \{\mathcal{C}^\alpha\}_{\alpha \in \mathcal{A}})$ is equivalent to $(\mathcal{P}, \mathcal{C})$ as they have identical first-stage cost and worst-case second-stage cost. It can also be seen that the flat solution $(\mathcal{P}, \{\mathcal{C}^\alpha\}_{\alpha \in \mathcal{A}})$ is still feasible if we impose on an additional constraint that the total number of consolidations on each arc $\alpha \in \mathcal{A}$ cannot exceed $|\mathcal{C}^\alpha|$ in the first stage of the problem. This enlightens a novel idea to enhance the C&G method that we can dynamically adjust the maximum total number of consolidations allowed on each arc to restrict the sizes of master problems and the subproblems, so as to accelerate the C&CG method.

By following the idea above, we propose an enhanced column-and-constraint generation method (EC&CG) in this section. It introduces a parameter $\mathbf{\Pi}$ to specify the maximum total number of consolidations allowed on each arc, the value of which is small at start, and then iteratively adjusted, together with the update of the scenario subset Λ. As a result, both the master problem and the subproblems solved in the EC&CG method have small sizes during the early iterations, which accelerates the search for the upper bound of the optimal objective value of the problem. However, it postpones the verification of the optimality of the upper bound to the later iterations, where it requires to set a sufficiently large value for the parameter $\mathbf{\Pi}$.

In the followings, we first introduce the parameter $\mathbf{\Pi}$ in Section 3.5.1 and propose the framework of the EC&CG method which dynamically adjusts the value of the parameter $\mathbf{\Pi}$ in Section 3.5.2. To illustrate how the parameter $\mathbf{\Pi}$ is adjusted in detail, we first derive an upper bound for $\mathbf{\Pi}$ in Section 3.5.3. Based on this upper bound, we then design our strategies to initialize and adjust the value of $\mathbf{\Pi}$ in Sections 3.5.4 and

3.5.5, respectively. Finally, we prove that our EC&CG method converge to an optimal solution to the robust CTSNDP-HC in a finite number of iterations in Section 3.5.6.

3.5.1 Parameterizing the Two-Stage MINLP Model

Model RO proposed in Section 3.3.2 is based on the fact that there are at most $|\mathcal{K}|$ consolidations on each arc. We now introduce a parameter (vector) $\mathbf{\Pi} = \{\Pi_{ij}\}_{(i,j)\in\mathcal{A}}$ referred to as the *consolidation frequency bound* to replace the terms $|\mathcal{K}|$ in model RO. This parameter $\mathbf{\Pi}$ restricts that the total number of consolidations on each arc $(i,j) \in \mathcal{A}$ in any feasible solution of the robust CTSNDP-HC cannot exceed Π_{ij} of $\mathbf{\Pi}$. Let $\mathcal{X}(\mathbf{\Pi})$ denote the domain defined by (3.27)–(3.35) under the consolidation frequency bound $\mathbf{\Pi}$, i.e., with $|\mathcal{K}|$ in constraints (3.27)–(3.35) replaced by the corresponding Π_{ij} for each $(i,j) \in \mathcal{A}$. We can now extend model RO to define the following model RO($\mathbf{\Pi}$) for every given parameter $\mathbf{\Pi}$:

$$[\mathrm{RO}(\mathbf{\Pi})] \qquad \min_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\overline{\boldsymbol{v}},\overline{\boldsymbol{b}})\in\mathcal{X}(\mathbf{\Pi})} \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{\Pi_{ij}} f_{ij} \cdot y_{ijr} + F_{RP}(\boldsymbol{x},\boldsymbol{z},\mathbf{\Pi}) \tag{3.78}$$

where $F_{RP}(\boldsymbol{x},\boldsymbol{z},\mathbf{\Pi})$ indicates the worst-case second-stage cost for the first-stage solution $(\boldsymbol{x},\boldsymbol{z})$ under $\mathbf{\Pi}$. We can also extend model SO($\boldsymbol{x},\boldsymbol{z}$) to define model SO($\boldsymbol{x},\boldsymbol{z},\mathbf{\Pi}$) by substituting terms $|\mathcal{K}|$ with their corresponding Π_{ij} for each $(i,j) \in \mathcal{A}$, so that $F_{RP}(\boldsymbol{x},\boldsymbol{z},\mathbf{\Pi})$ equals the optimal objective value of SO($\boldsymbol{x},\boldsymbol{z},\mathbf{\Pi}$) . Model RO($\mathbf{\Pi}$) aims to find the optimal robust solution among all the solutions in which the total number of consolidations on each arc $(i,j) \in \mathcal{A}$ is not greater than Π_{ij}.

Let $\mathbf{\Pi}^{\mathcal{K}}$ indicate a special consolidation frequency bound such that $\Pi_{ij}^{\mathcal{K}} = |\mathcal{K}|$, for all $(i,j) \in \mathcal{A}$. It can be seen that model RO proposed in Section 3.3.2 is equivalent to RO($\mathbf{\Pi}^{\mathcal{K}}$).

3.5.2 Algorithm Framework

The master problem and the subproblem of our EC&CG method can be formulated as follows. Let $\mathcal{Q}(\tilde{\boldsymbol{\tau}}, \boldsymbol{\Pi})$ denote the domain of variables $(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{s})$ defined by (3.37)–(3.47) under a given travel time realization $\tilde{\boldsymbol{\tau}}$ and the consolidation frequency bound $\boldsymbol{\Pi}$, where all terms $|\mathcal{K}|$ in (3.37)–(3.47) are replaced with their corresponding Π_{ij} for each $(i,j) \in \mathcal{A}$. Accordingly, we can obtain the the master problem to be solved during each iteration n of the EC&CG method, which is denoted by RMP$(\boldsymbol{\Pi}^{(n)}, \Lambda^{(n)})$ with $\boldsymbol{\Pi}^{(n)} \leq \boldsymbol{\Pi}^{\mathcal{K}}$ and $\Lambda^{(n)} \in \mathbb{U}(\Gamma)$, as follows:

[RMP$(\boldsymbol{\Pi}^{(n)}, \Lambda^{(n)})$]

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{\Pi_{ij}^{(n)}} f_{ij} \cdot y_{ijr} + \phi \tag{3.79}$$

$$s.t. \quad \phi \geq \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} (h_i^k q^k) \cdot w_i^{k(\tilde{\boldsymbol{\tau}})} + \sum_{k \in \mathcal{K}} g^k \cdot s^{k(\tilde{\boldsymbol{\tau}})}, \quad \forall\, \tilde{\boldsymbol{\tau}} \in \Lambda^{(n)}, \tag{3.80}$$

$$(\boldsymbol{v}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{b}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{w}^{(\tilde{\boldsymbol{\tau}})}, \boldsymbol{s}^{(\tilde{\boldsymbol{\tau}})}) \in \mathcal{Q}(\tilde{\boldsymbol{\tau}}, \boldsymbol{\Pi}^{(n)}), \quad \forall\, \tilde{\boldsymbol{\tau}} \in \Lambda^{(n)}, \tag{3.81}$$

$$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{b}}) \in \mathcal{X}(\boldsymbol{\Pi}^{(n)}). \tag{3.82}$$

Consider the master problem RMP$(\boldsymbol{\Pi}^{(n)}, \Lambda^{(n)})$ in iteration n, with its optimal solution denoted as $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)})$. We can obtain the formulation of the subproblem, denoted by SRP$(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Pi})$, by substituting terms $|\mathcal{K}|$ with corresponding Π_{ij} in model SRP$(\boldsymbol{x}, \boldsymbol{z})$ for each $(i,j) \in \mathcal{A}$. Thus, model SO$(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Pi})$ can be reformulated as model SRP$(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Pi})$, by following the reformulation method used in Section 3.4.1. The worst-case second-stage cost $F_{RP}(\boldsymbol{x}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{\Pi}^{(n)})$ can thus be obtained by solving the subproblem SRP$(\boldsymbol{x}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{\Pi}^{(n)})$. Based on the solution $\boldsymbol{\zeta}^{(n)}$ obtained from model SRP$(\boldsymbol{x}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{\Pi}^{(n)})$, the worst-case scenario $\tilde{\boldsymbol{\tau}}^{(n)} \in \mathbb{U}(\Gamma)$ can be identified according to (3.83) below and then be used to update $\Lambda^{(n+1)}$.

$$\tilde{\tau}_{ijr}^{(n)} = \begin{cases} \sum\limits_{\ell \in \{-1,0,1\}} \tilde{\tau}_{ijr,\ell} \zeta_{ijr,\ell}^{(n)}, & \text{if } r \in \{1, ..., \Pi_{ij}\}, \\ \overline{\tau}_{ijr}, & \text{otherwise,} \end{cases} \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}. \tag{3.83}$$

Our EC&CG method starts from a small $\mathbf{\Pi}$ and empty Λ, iteratively solves the master problem and subproblem, and dynamically updates $\mathbf{\Pi}$ as well as Λ, until a certain termination condition is reached. The framework of the EC&CG method can be summarized as follows: Given an initial value $\mathbf{\Pi}^{(1)}$ of the consolidation frequency bound, it sets $\Lambda^{(1)} = \emptyset$, UB = ∞, and LB = -∞, and then in each iteration $n = 1, 2, ...$, it goes through the following steps:

Step 1: Solve the master problem RMP($\mathbf{\Pi}^{(n)}, \Lambda^{(n)}$) defined by $\mathbf{\Pi}^{(n)}$ and $\Lambda^{(n)}$ to find its optimal solution $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \phi^{(n)})$, and let LB equal to the optimal objective value of the RMP($\mathbf{\Pi}^{(n)}, \Lambda^{(n)}$).

Step 2: Solve the subproblem SRP($\boldsymbol{x}^{(n)}, \boldsymbol{z}^{(n)}, \mathbf{\Pi}^{(n)}$) optimally to obtain the worst-case second-stage cost $F_{RP}(\boldsymbol{x}^{(n)}, \boldsymbol{z}^{(n)}, \mathbf{\Pi}^{(n)})$, identify the worst-case scenario $\tilde{\boldsymbol{\tau}}^{(n)} \in \mathbb{U}(\Gamma)$ according to (3.83), and update

$$\Lambda^{(n+1)} = \Lambda^{(n)} \bigcup \{\tilde{\boldsymbol{\tau}}^{(n)}\}, \tag{3.84}$$

$$UB = \min\{UB, \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^{k(n)} + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{\Pi_{ij}^{(n)}} f_{ij} \cdot y_{ijr}^{(n)} + F_{RP}(\boldsymbol{x}^{(n)}, \boldsymbol{z}^{(n)}, \mathbf{\Pi}^{(n)})\}. \tag{3.85}$$

Step 3: If a certain stopping condition is reached, then the algorithm stops. Otherwise, adjust the value of the consolidation frequency bound with the new value denoted by $\mathbf{\Pi}^{(n+1)}$. Update $n = n + 1$ and then go to Step 2.

To apply our EC&CG method to solving the robust CTSNDP-HC under uncertain travel times, we further design some specific strategies in Section 3.5.4 and Section 3.5.5

to determine the initial value of the consolidation frequency bound at beginning, and to adjust the value of the consolidation frequency bound in each iteration, respectively. We also specify the stopping condition for Step 3 and discuss the convergence of the EC&CG method in Section 3.5.6. For these, we first derive a bound on the value of the consolidation frequency bound, which is smaller than $\mathbf{\Pi}^{\mathcal{K}}$ but is large enough to prove the optimality of the obtained solution to the original robust CTSNDP-HC, in Section 3.5.3.

3.5.3 Bounding Parameter Values

Under the constraints imposed by the commodities' earliest available times and due times, the total number of consolidations on an arc can be strictly less than $|\mathcal{K}|$ in every nominal timely-implementable flat solution. For example, consider any commodity $k \in \mathcal{K}$ and arc $(i,j) \in \mathcal{A}$. If the total travel time of every path from the origin o^k to the destination d^k of commodity k that passes the arc (i,j) exceeds $(l^k - e^k)$ (i.e., the the difference of the due time l^k and the earliest available time e^k of commodity k), then commodity k cannot pass arc (i,j) without violating its due time, implying that the total number of consolidations on arc (i,j) must be less than or equal $|\mathcal{K}| - 1$.

Based on the observation above, we define Π^*_{ij} for each arc $(i,j) \in \mathcal{A}$ as the total number of commodities k such that $\underline{\tau}^k(o^k, i) + \overline{\tau}_{ij} + \underline{\tau}^k(j, d^k) \leq l^k - e^k$, where $\underline{\tau}^k(i', j')$ for each pair of nodes i' and j' indicates the length of the shortest-time path from node i' to node j' for commodity k under the nominal travel times in the flat network. The argument for our observation above indicates that only such commodity k can pass arc (i,j) without violating its due time. Thus, the total number of consolidations on arc (i,j) must be less than or equal to Π^*_{ij}. Therefore, we refer to the vector $\mathbf{\Pi}^*$ of Π^*_{ij} for $(i,j) \in \mathcal{A}$ as the *maximum consolidation frequency.* We have that $\mathbf{\Pi}^* \leq \mathbf{\Pi}^{\mathcal{K}}$ because $\Pi^*_{ij} \leq |\mathcal{K}|$ for all $(i,j) \in \mathcal{A}$.

We can now establish the following Proposition 4 which indicates that the optimal objective values of models RO($\mathbf{\Pi}^*$) and RO($\mathbf{\Pi}^{\mathcal{K}}$) are equal.

Proposition 4. *RO*($\mathbf{\Pi}^{\mathcal{K}}$) *and RO*($\mathbf{\Pi}^{*}$) *have the same optimal objective values.*

Proof. Given an optimal solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ to RO($\mathbf{\Pi}^{\mathcal{K}}$) with the objective value

$$Z_1 = \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k q^k) \cdot x_{ij}^k + \sum_{(i,j)\in\mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} \cdot y_{ijr} + F_{RP}(\boldsymbol{x}, \boldsymbol{z}),$$

where $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ indicates the worst-case second-stage cost and equals the optimal objective value of model SO$(\boldsymbol{x}, \boldsymbol{z})$. From $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, we can derive a flat solution $(\mathcal{P}(\boldsymbol{x}, \boldsymbol{z}), \mathcal{C}(\boldsymbol{x}, \boldsymbol{z}))$. Let $\mathcal{R}^{\alpha}(\boldsymbol{x}, \boldsymbol{z}) = \{r : C_r^{\alpha}(\boldsymbol{x}, \boldsymbol{z}) \neq \emptyset\}$ indicate a set of non-empty consolidations on each arc $\alpha \in \mathcal{A}$. By the definition of $\mathbf{\Pi}^{*}$, we have $|\mathcal{R}^{\alpha}(\boldsymbol{x}, \boldsymbol{z})| \leq \mathbf{\Pi}_{\alpha}^{*}$, $\forall \alpha \in \mathcal{A}$. Accordingly, by re-indexing the non-empty consolidations in each $C^{\alpha}(\boldsymbol{x}, \boldsymbol{z})$, we can derive a feasible solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ for RO($\mathbf{\Pi}^{\mathcal{K}}$) from $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ where

$$\begin{aligned}
&\hat{\boldsymbol{x}} = \boldsymbol{x}, \\
&\hat{y}_{ijr} = \begin{cases} y_{ij,\mathcal{R}_r^{(i,j)}(\boldsymbol{x},\boldsymbol{z})}, & \text{if } r \in \{1, ..., |\mathcal{R}^{(i,j)}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})|\}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}, \\
&\hat{z}_{ijr}^k = \begin{cases} z^k_{ij,\mathcal{R}_r^{(i,j)}(\hat{\boldsymbol{x}},\hat{\boldsymbol{z}})}, & \text{if } r \in \{1, ..., |\mathcal{R}^{(i,j)}(\boldsymbol{x}, \boldsymbol{z})|\}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., |\mathcal{K}|\}.
\end{aligned}$$

It can be seen that re-indexing the consolidations does not change the actual execution of the routing and consolidation plans, and does not change the corresponding worst-case second-stage cost. Thus, since $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is an optimal solution to model RO($\mathbf{\Pi}^{\mathcal{K}}$), we obtain that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ is also an optimal solution to model RO($\mathbf{\Pi}^{\mathcal{K}}$).

Therefore, by adding the following constraints (3.86) to model RO($\mathbf{\Pi}^{\mathcal{K}}$), it does not change the optimal objective value.

$$z_{ijr}^k = 0, \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \{\Pi_{ij}^* + 1, ..., |\mathcal{K}|\} \tag{3.86}$$

By fixing such variables z_{ijr}^k for $(i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \{\Pi_{ij}^* + 1, ..., |\mathcal{K}|\}$ to be zero, as imposed by the new constraints (3.86), it is equivalent to removing these variables from model RO($\mathbf{\Pi}^{\mathcal{K}}$), which exactly leads to model RO($\mathbf{\Pi}^*$). Hence, models RO($\mathbf{\Pi}^{\mathcal{K}}$) and RO($\mathbf{\Pi}^*$) must have the same optimal objective values. □

Proposition 4 indicates that for each arc $(i, j) \in \mathcal{A}$, the value of Π_{ij} can be limited to no greater than Π^*_{ij}. Therefore, when we set the initial value and adjust the value of the parameter Π_{ij} for each arc $(i, j) \in \mathcal{A}$, we can take into account only values no greater than Π^*_{ij}. When $\mathbf{\Pi}$ reaches the value of $\mathbf{\Pi}^*$, the EC&CG method becomes the standard C&CG method, and it does not need to update the value of $\mathbf{\Pi}$ any more.

As we will show later, in each iteration n of the EC&CG method, the UB obtained is a valid upper bound for both model RO($\mathbf{\Pi}^{(n)}$) and the original robust CTSNDP-HC. In each iteration n of the EC&CG method, the LB obtained is a valid lower bound for model RO($\mathbf{\Pi}^{(n)}$) since RMP($\mathbf{\Pi}^{(n)}, \Lambda^{(n)}$) is a relaxation of model RO($\mathbf{\Pi}^{(n)}$). However, the LB obtained may not be a valid lower bound for the original robust CTSNDP-HC since the solution space of model RO($\mathbf{\Pi}^{(n)}$) is restricted by the parameter $\mathbf{\Pi}^{(n)}$. Only when $\Pi^{(n)}_{ij}$ equals Π^*_{ij} for each $(i, j) \in \mathcal{A}$, the LB obtained becomes a valid lower bound for the original robust CTSNDP-HC.

3.5.4 Initialization of Parameter Values

It is critical for the EC&CG method to set the initial consolidation frequency bound $\mathbf{\Pi}^{(1)}$ properly. On the one hand, it should ensure the feasibility of the RMP($\mathbf{\Pi}^{(1)}, \Lambda^{(1)}$) and can produce a reasonably good initial upper bound. On the other hand, it should also guarantee the tractability of the RMP($\mathbf{\Pi}^{(1)}, \Lambda^{(1)}$).

To determine $\mathbf{\Pi}^{(1)}$, we first follow the two steps below to generate an initial feasible first-stage solution that has relatively few consolidations on each arc.

Step 1: For each commodity $k \in \mathcal{K}$, compute a feasible delivery path P^k that does not violate the due time under the nominal travel times, with its total transportation cost (including flow costs and fixed costs) minimized. For each arc $(i, j) \in \mathcal{A}$, let $\hat{\Pi}_{ij}$ denote the total number of times that arc (i, j) has appeared in the obtained paths P^k.

Step 2: Solve the deterministic model DO with x_{ij}^k fixed to be one if $(i, j) \in P^k$, with x_{ij}^k fixed to be zero if $(i, j) \notin P^k$, and with terms $|\mathcal{K}|$ replaced by $\hat{\Pi}_{ij}$ for all $(i, j) \in \mathcal{A}$ and $k \in \mathcal{K}$. The solution obtained is adopted as the initial first-stage solution, denoted by $(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$.

The Step 1 above constructs a routing plan for all the commodities by following a greedy approach with respect to the transportation cost. The Step 2 above then optimizes the consolidation plan based on the routing plan obtained in Step 1, which is expected to result in a relatively small number of consolidations on each arc. Let $\mathcal{C}^{\alpha}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$ denote the set of non-empty consolidations on each arc $\alpha \in \mathcal{A}$ in solution $(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$. For each arc $(i, j) \in \mathcal{A}$, by the definition of Π_{ij}^* we know that $|\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})| \leq \Pi_{ij}^*$. Let ρ_{ij}^0 indicate a threshold given as a parameter. Accordingly, we determine the value $\Pi_{ij}^{(1)}$ as follows. If $|\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})| \geq \rho_{ij}^0$, then we set $\Pi_{ij}^{(1)} = |\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})|$, and otherwise, we set $\Pi_{ij}^{(1)} = \min\{\rho_{ij}^0, \Pi_{ij}^*\}$. In other words, we set $\Pi_{ij}^{(1)} = \max\Big\{\min\{\rho_{ij}^0, \Pi_{ij}^*\}, |\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})|\Big\}$ for each $(i, j) \in \mathcal{A}$. It can be seen that $\Pi_{ij}^{(1)} \geq |\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})|$ for all $(i, j) \in \mathcal{A}$, ensuring that model RMP$(\boldsymbol{\Pi}^{(1)}, \Lambda^{(1)})$ at least contains $(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$ as its feasible solution and can provide an upper bound better than the worst-case total cost of the solution $(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$. The value of $\Pi_{ij}^{(1)}$ is set to be $|\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})|$ or $\min\{\rho_{ij}^0, \Pi_{ij}^*\}$, whichever is larger, so that the tractability of the model RMP$(\boldsymbol{\Pi}^{(1)}, \Lambda^{(1)})$ can be ensured when ρ_{ij}^0 is small, as the value of $|\mathcal{C}^{(i,j)}(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})|$ is expected to be small due to Step 2 above. In our computations study in Section 3.7, we choose $\rho_{ij}^0 = 4$, for all $(i, j) \in \mathcal{A}$.

3.5.5 Adjustment of Parameter Values

At each iteration of the EC&CG method, the consolidation frequency bound $\boldsymbol{\Pi}$ needs to be increased, so as to expand the solution space of the master problem. While there may be different strategies to adjust the consolidation frequency bound, we propose one described below.

During each iteration $n = 1, 2, ...,$ of the EC&CG method, after LB and UB are updated, if the algorithm does not stop, we need to set the values of $\Pi_{ij}^{(n+1)}$ for $(i, j) \in \mathcal{A}$

to update the consolidation frequency bound. When $\Pi_{ij}^{(n)} = \Pi_{ij}^{*}$ for all $(i,j) \in \mathcal{A}$, we keep $\Pi_{ij}^{(n+1)} = \Pi_{ij}^{*}$ for all $(i,j) \in \mathcal{A}$, so that the standard C&CG method for RO($\boldsymbol{\Pi}^{*}$) is followed in the future iterations. If $\Pi_{ij}^{(n)} < \Pi_{ij}^{*}$ for some $(i,j) \in \mathcal{A}$, we set the values of $\Pi_{ij}^{(n+1)}$ for $(i,j) \in \mathcal{A}$ according to the following two possible cases:

- Case 1: If the optimality $(UB - LB)/UB$ is greater than 0.01, then we first set $\Pi_{ij}^{(n+1)}$ to be equal to $\Pi_{ij}^{(n)}$ for each arc $(i,j) \in \mathcal{A}$. We then compute ρ_{ij} that indicates the total number of times that arc (i,j) has appeared in all the commodities' delivery paths of the solution $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \phi^{(n)})$ for all arc $(i,j) \in \mathcal{A}$. If $\rho_{ij} > \Pi_{ij}^{(n+1)}$, we increase $\Pi_{ij}^{(n+1)}$ to ρ_{ij}, so as to allow more consolidations on such arcs (i,j) in future iterations. To extend the adjustment strategy to the arcs not involved in the obtained solution $(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \phi^{(n)})$ so as to further expand the solution space in the future iterations, we increase $\Pi_{ij}^{(n+1)}$ by one for the arc (i,j) with the least value of $\Pi_{ij}^{(n+1)}$ that is smaller than Π_{ij}^{*}. This also ensures that $\Pi_{ij}^{(n+1)}$ is greater than $\Pi_{ij}^{(n)}$ for at least one arc (i,j).

- Case 2: If the optimality $(UB - LB)/UB$ is smaller or equal to 0.01, then we set $\Pi_{ij}^{(n)} = \Pi_{ij}^{*}$ for all $(i,j) \in \mathcal{A}$, so that further iterations of the EC&CG method follows the standard C&CG method for RO($\boldsymbol{\Pi}^{*}$) to produce valid lower bounds LB, which can then be used to verify the optimality of the newly obtained upper bounds UB.

The above adjustment strategy implies that after some iteration n, we always have $\Pi_{ij}^{(n)} = \Pi_{ij}^{*}$ for all $(i,j) \in \mathcal{A}$. This ensures that LB obtained in each iteration after iteration n is always a valid lower bound for the original robust CTSNDP-HC, so that it can be shown that the EC&CG method converges to an optimal solution to the original robust CTSNDP-HC (see Section 3.5.6 below). This adjustment process can indeed be seen as a simple learning process that learns the consolidation information from previous obtained solutions.

3.5.6 Stopping Condition and Convergence Guarantee

For the EC&CG method, a natural stopping condition is that $\Pi_{ij}^{(n)} = \Pi_{ij}^{*}$ for all $(i,j) \in \mathcal{A}$ and the optimality gap $(UB - LB)/UB = 0$. It can be shown as follows that under such a stopping condition, the EC&CG method must stop and converge to the an optimal solution to the robust CTSNDP-HC in a finite number of iterations.

First, we need to establish Proposition 5 below.

Proposition 5. *The value of UB obtained in each iteration n of the EC&CG method is a valid upper bound on both the optimal objective value of RO($\mathbf{\Pi}^{(n)}$) and the optimal objective value of RO($\mathbf{\Pi}^{*}$).*

Proof. Let $Z^{(n)}$ represent the optimal objective value of RO($\mathbf{\Pi}^{(n)}$). If we can prove that for each iteration n' of the EC&CG method with $1 \leq n' \leq n-1$, every feasible solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ to RO($\mathbf{\Pi}^{(n')}$), of which the objective value is denoted by $Z^{(n')}$, satisfies that $Z^{(n')} \geq Z^{(n)}$, then from (3.85), we can obtain that the value of UB obtained in each iteration n is a valid upper bound on $Z^{(n)}$.

By definition, $Z^{(n')}$ can be represented as follows.

$$Z^{(n')} = \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c_{ij}^{k} q^{k}) \cdot x_{ij}^{k} + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{\Pi_{ij}^{(n')}} f_{ij} \cdot y_{ijr} + F_{RP}(\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Pi}^{(n')}),$$

where $F_{RP}(\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Pi}^{(n')})$ denotes the worst-case second-stage cost of the first-stage solution $(\boldsymbol{x}, \boldsymbol{z})$ under $\mathbf{\Pi}^{(n')}$, and it equals the optimal objective value of model SO($\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Pi}^{(n')}$). From solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, we can derive a feasible solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ to RO($\mathbf{\Pi}^{(n)}$) as follows:

$$\hat{\boldsymbol{x}} = \boldsymbol{x},$$

$$\hat{y}_{ijr} = \begin{cases} y_{ijr}, & \text{if } r \in \{1, ..., \Pi_{ij}^{(n')}\}, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., \Pi_{ij}^{(n)}\},$$

$$\hat{z}_{ijr}^{k} = \begin{cases} z_{ijr}^{k}, & \text{if } r \in \{1, ..., \Pi_{ij}^{(n')}\}, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall\ (i,j) \in \mathcal{A}, r \in \{1, ..., \Pi_{ij}^{(n)}\}.$$

The objective value of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, denoted by $\hat{Z}$, can be represented as follows:

$$\hat{Z} = \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c_{ij}^k q^k) \cdot \hat{x}_{ij}^k + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{\Pi_{ij}^{(n)}} f_{ij} \cdot \hat{y}_{ijr} + F_{RP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \boldsymbol{\Pi}^{(n)}),$$

and satisfies that $\hat{Z} \geq Z^{(n)}$.

It can also be seen that $\mathrm{SO}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \boldsymbol{\Pi}^{(n)})$ can be transformed equivalently to $\mathrm{SO}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Pi}^{(n')})$ by removing those redundant variables and constraints that are associated with $\hat{z}_{ijr}^k$ having $r > \Pi_{ij}^{(n')}$. Thus, $F_{RP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \boldsymbol{\Pi}^{(n)})$ must be equal to $F_{RP}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\Pi}^{(n')})$, implying that $\hat{Z} = Z^{(n')}$. Accordingly, we have $Z^{(n')} = \hat{Z} \geq Z^{(n)}$, which, together with (3.85), implies that the value of UB obtained in each iteration n is an upper bound on $Z^{(n)}$.

Moreover, since $Z^{(n)}$ is less than or equal to the optimal objective value of $\mathrm{RO}(\boldsymbol{\Pi}^*)$, we obtain the value of UB must also be an upper bound on the optimal objective value of $\mathrm{RO}(\boldsymbol{\Pi}^*)$. This completes the proof of Proposition 5. $\square$

From Proposition 5, we know that in each iteration n of the EC&CG method, the value of UB is always a valid upper bound for $\mathrm{RO}(\boldsymbol{\Pi}^{(n)})$. Thus, similar to our argument for the C&CG method in Section 3.4.2, we can see that after a finite number of iterations with the same value of $\Pi_{ij}^{(n)}$, the gap $(UB - LB)/UB$ can reach zero. Thus, according to the adjustment strategies proposed in Section 3.5.5 for the consolidation frequency bound $\boldsymbol{\Pi}$, if there exists $\Pi_{ij}^{(n)}$ that is less than Π_{ij}^*, then at least one such $\Pi_{ij}^{(n)}$ must be increased by at least one in each iteration. Since each $\Pi_{ij}^{(n)}$ cannot exceed Π_{ij}^*, we can see that each $\Pi_{ij}^{(n)}$ must reach Π_{ij}^* after a finite number of iterations. This implies that after a finite number of iterations, the EC&CG method must eventually follow the standard C&CG method to solve the $\mathrm{RO}(\boldsymbol{\Pi}^*)$. Thus, from our argument in Section 3.4.2 for the standard C&CG method, we obtain that our C&CG method can satisfy the stopping condition $\Pi_{ij}^{(n)} = \Pi_{ij}^*$ for all $(i,j) \in \mathcal{A}$ and the optimality gap $(UB - LB)/UB = 0$, and thus, converge to an optimal solution to the original robust CTSNDP-HC, in a finite number of iterations.

Moreover, in some situations, one may impose a time limit stopping condition such that the EC&CG method stops when a maximum running time allowed is exceeded. Under this stopping condition, our EC&CG method may stop in certain iteration n with $\Pi_{ij}^{(n)} < \Pi_{ij}^{*}$ for some $(i, j) \in \mathcal{A}$. It is worthy to note that in such a situation, the LB obtained is a valid lower bound only for RO($\mathbf{\Pi}^{(n)}$) but may not be a valid lower bound for RO($\mathbf{\Pi}^{*}$). This is because when such a smaller consolidation frequency bound $\mathbf{\Pi}^{(n)}$ is applied, some feasible solutions to RO($\mathbf{\Pi}^{*}$) may be ruled out in RO($\mathbf{\Pi}^{(n)}$).

However, even for such a situation caused by the time limit stopping condition, where our EC&CG method stops with $\Pi_{ij}^{(n)} < \Pi_{ij}^{*}$ for some $(i, j) \in \mathcal{A}$, Proposition 5 implies that the value of UB obtained is a valid upper bound on the optimal objective value of the original robust CTSNDP-HC.

Moreover, likewise, this EC&CG method can also be extended to solve the cases with non-integer budget of uncertainty Γ by slightly modifying the Proposition 3 and the cases with strict robustness by using an infinity large delay penalty $\boldsymbol{g}$.

3.6 Additional Accelerating Strategies

To further enhance the performance of our newly proposed C&CG and EC&CG methods, we develop and apply two additional accelerating strategies, which are illustrated in Section 3.6.1 and Section 3.6.2, respectively.

3.6.1 Size Reduction for the Subproblem

The first strategy is to further reduce the size of the subproblem. In the followings, we first present the size reduction for the subproblem SRP($\boldsymbol{x}, \boldsymbol{z}$) of the C&CG method, and then show that the approach can be applied to the subproblem SRP($\boldsymbol{x}, \boldsymbol{z}, \mathbf{\Pi}$) of the EC&CG method.

To break the symmetric structure of each master problem $\text{MP}_{C\&CG}$ of the C&CG method, we can restrict that consolidation r on each arc $(i, j) \in \mathcal{A}$ is not empty only if consolidation $r - 1$ is not empty. As a result, we can impose the following cut on the first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ in the master problem RMP$(\boldsymbol{\Pi}^{(n)}, \Lambda^{(n)})$:

$$y_{ijr+1} \leq M y_{ijr}, \quad \forall\ (i,j) \in \mathcal{A}, r \in \{1,2,...,|\mathcal{K}|-1\}, \tag{3.87}$$

where M is a sufficiently large number.

Consider any first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ with (3.87) satisfied. We can reduce the size of the subproblem SRP$(\boldsymbol{x}, \boldsymbol{z})$ as follows. First, from $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, we can obtain its flat solution $(\mathcal{P}(\boldsymbol{x}, \boldsymbol{z}), \mathcal{C}(\boldsymbol{x}, \boldsymbol{z}))$. Let $P^k(\boldsymbol{x}, \boldsymbol{z})$ denote the 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 the node sequence and the arc sequence along $P^k(\boldsymbol{x}, \boldsymbol{z})$, respectively. Let $\mathcal{C}^\alpha(\boldsymbol{x}, \boldsymbol{z})$ denote the set that consists of all non-empty consolidations on arc $\alpha \in \mathcal{A}$. Accordingly, there are totally $|\mathcal{C}^\alpha(\boldsymbol{x}, \boldsymbol{z})|$ consolidations on each arc $\alpha \in \mathcal{A}$.

Second, in any optimal solution to model SO$(\boldsymbol{x}, \boldsymbol{z})$ of the second-stage problem, it can be seen that $v_{ij}^k = 0$ for all $(i,j) \notin \mathcal{A}^k$, $k \in \mathcal{K}$, $w_i^k = 0$ for all $i \notin \mathcal{N}^k$, $k \in \mathcal{K}$, and $b_{ijr} = 0$ for all $(i,j) \in \mathcal{A}$ with $C_r^{(i,j)}(\boldsymbol{x}, \boldsymbol{z})$ being empty. Thus, these variables do not affect the optimal objective value $F_{RP}(\boldsymbol{x}, \boldsymbol{z})$ of model SO$(\boldsymbol{x}, \boldsymbol{z})$, leading to a number of redundant variables and constraints in model SO$(\boldsymbol{x}, \boldsymbol{z})$, which can thus be excluded. Accordingly, we can reduce the size of model SO$(\boldsymbol{x}, \boldsymbol{z})$ by replacing terms $\mathcal{N}$, $\mathcal{A}$ and Π_{ij} with 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.

Third, with the size of model SO$(\boldsymbol{x}, \boldsymbol{z})$ reduced, the size of its reformulation, model SRP$(\boldsymbol{x}, \boldsymbol{z})$, can also be reduced. The resulting model, denoted by SRP$_1(\boldsymbol{x}, \boldsymbol{z})$, is shown as follows, where $\mathcal{L}(\boldsymbol{x}, \boldsymbol{z})$ denote the domain defined by (3.53) – (3.60), (3.66) – (3.68) and (3.69) with terms $\mathcal{N}$, $\mathcal{A}$ and $|\mathcal{K}|$ replaced by 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.

$$\begin{aligned}[\text{SRP}_1(\boldsymbol{x}, \boldsymbol{z})]\ F_{RP}(\boldsymbol{x}, \boldsymbol{z}) = \max_{(\boldsymbol{\zeta},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\psi},\boldsymbol{\eta},\boldsymbol{\theta},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\varphi}) \in \mathcal{L}(\boldsymbol{x}, \boldsymbol{z})} & \sum_{(j,i) \in \mathcal{A}^k} \sum_{r=1}^{|\mathcal{C}^{(j,i)}(\boldsymbol{x}, \boldsymbol{z})|} \varphi_{jir} \\ & - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}^k} (M_1 x_{ij}^k) \cdot \eta_{ij}^k\end{aligned}$$

$$+\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{A}^k}\sum_{r=1}^{|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|}[M_1(z_{ijr}^k-1)]\cdot(\theta_{ijr}^k+\xi_{ijr}^k)$$
$$+\sum_{k\in\mathcal{K}}e^k\cdot(\gamma^k-\lambda_{o^k}^k)+\sum_{k\in\mathcal{K}}l^k\cdot(\lambda_{d^k}^k-\psi^k) \tag{3.88}$$

Therefore, for the subproblem of the C&CG method, to compute the worst-case second-stage cost $F_{RP}(\boldsymbol{x},\boldsymbol{z})$, we only need to solve model $\mathrm{SRP}_1(\boldsymbol{x},\boldsymbol{z})$, in which the consolidation index r of each arc $(i,j)\in\mathcal{A}$ is bounded by $|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|$ instead of $|\mathcal{K}|$.

For the EC&CG method, to break the symmetric structure of each master problem $\mathrm{RMP}(\boldsymbol{\Pi},\Lambda)$, we can impose a constraint similar to (3.87) with $|\mathcal{K}|$ replaced by Π_{ij}. For any first-stage solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ with such a constraint satisfied, we can follow an argument similar to the above for the C&CG method to show that for the subproblem of the EC&CG method, to compute the optimal objective value $F_{RP}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\Pi})$, we also only need to solve model $\mathrm{SRP}_1(\boldsymbol{x},\boldsymbol{z})$.

With the size of the subproblem reduced, the worst-case scenario $\tilde{\boldsymbol{\tau}}\in\mathbb{U}(\Gamma)$ for each first-stage solution $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ can be identified according to (3.89) below, so that the convergences of the C&CG and EC&CG methods are still guaranteed.

$$\tilde{\tau}_{ijr}=\begin{cases}\sum_{\ell\in\{-1,0,1\}}\tilde{\tau}_{ijr,\ell}\zeta_{ijr,\ell}, & \text{if } r\in\{1,...,|\mathcal{C}^{(i,j)}(\boldsymbol{x},\boldsymbol{z})|\},\\ \overline{\tau}_{ijr}, & \text{otherwise},\end{cases}\quad\forall\ (i,j)\in\mathcal{A},r\in\{1,...,|\mathcal{K}|\}. \tag{3.89}$$

3.6.2 Bundle of Worst-Case Scenarios

For both the C&CG and EC&CG methods, we can further accelerate them by applying a bundle strategy to update the scenario set and the corresponding cuts, which is similar to the strategy used by Remli et al. [87] to update cuts for the Benders decomposition method.

In each iteration of the C&CG and EC&CG methods, we solve the master problem to obtain its optimal first-stage solution and another pool of other feasible first-stage solutions, which can be achieved by common optimization solvers, such as CPLEX and Gurobi. These first-stage solutions are sorted in an non-decreasing order of their objective values.

For each first-stage solution obtained (including the optimal one) by the master problem, we then solve the corresponding subproblem to identify its worst-case scenario. Hence, in each iteration, there are multiple worst-case scenarios that can be identified, and whose corresponding new variables and constraints are available to be added to the master problem of the next iteration.

By adding the new variables and constraints obtained above, one can reduce the number of iterations required the convergence to the optimal solution, but may also increase the running time required to solve the master problem in each iteration. Thus, as a trade-off, we add only a bundle of at most two worst-case scenarios to the master problem in each iteration. One of such worst-case scenarios to add is that of the optimal first-stage solution. To identify another worst-case scenario to add, we evaluate the total first-stage and second-stage costs for all the first-stage solutions in the pool to update the value of UB, and choose to add the worst-case scenario of the first-stage solution in the pool that leads to an update of UB with the least total cost. (If no first-stage solution in the pool leads to an update of UB, only the worst-case scenario of the optimal first-stage solution is added.)

Moreover, when identifying the second worst-case scenario to add, we apply a bounding strategy similar to that in Lee et al. [77], so as to accelerate the evaluation of the worst-case total cost of each first-state solution in the pool. For this, consider each first-stage solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ in the pool, with its first-stage cost denoted by f_1. Accordingly, to evaluate the worst-case total cost of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, we need to solve model $\mathrm{SRP}_1(\boldsymbol{x}, \boldsymbol{z})$ to compute the worst-case second-stage cost of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. According to our bundle strategy described above, the worst-case scenario of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ can be added only if the worst-case total cost of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is less than UB. Thus, when using an optimization solver to solve model $\mathrm{SRP}_1(\boldsymbol{x}, \boldsymbol{z})$, we can enforce it to terminate earlier when it finds a lower bound on the second-stage cost of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ greater than $(\mathrm{UB} - f_1)$.

3.7 Computational Experiments

In this section, we present our computational experiments. We first evaluate the performance of the newly proposed C&CG and EC&CG methods in dealing with travel time uncertainty for instances of different scales. We then analyze the robustness and the price of the robustness of the newly proposed two-stage robust optimization model.

Both the C&CG and EC&CG methods were implemented in Java, with their corresponding master problems and subproblems solved by the solver of Gorubi (v.9.1.2). All experiments were conducted on a PC equipped with an Intel(R) Core(TM) i7-8700 CPU clocked at 3.20 GHz and 64 GB RAM, running on a 64-bit Windows 10 operating system.

3.7.1 Instances and Parameter Settings

We generated the instances for our computational experiments based on the 12 problem classes, R4-R15, of the fixed-charge capacitated multi-commodity network design problem in the literature [56]. These instances have been used to derive benchmark instances to evaluate the performance of the algorithms for the stochastic capacitated fixed charge network design problem under certain uncertainty [35, 91, 92]. The attributes of each class are given in Table 3.1, including the size of node set $|\mathcal{N}|$, the size of arc set $|\mathcal{A}|$, and the size of commodity set $|\mathcal{K}|$. Each class contains five networks indexed by 1, 3, 5, 7 and 9, indicating the corresponding "cost ratio" (the ratio of fixed cost to the variable cost) and "capacity ratio"(the ratio of total demand to total capacity of the network). These instances are referred to as "untimed" instances, since they do not have any temporal attributes, such as travel times of arcs, or earliest available and delivery due times of commodities.

We added time attributes to these instances by using a scheme similar to that in Boland et al. [20] to get the "timed" instances. First, for each arc $(i, j) \in \mathcal{A}$, we set $\overline{\tau}_{ij}$, the nominal value of travel time (in minutes), to be proportional to its fixed cost. According to Boland et al. [20], we set $\overline{\tau}_{ij} = 0.55 \times f_{ij}$ where f_{ij} represents

Table 3.1 Characteristics of “untimed” **R** instances

Class	$\vert\mathcal{N}\vert$	$\vert\mathcal{A}\vert$	$\vert\mathcal{K}\vert$	Class	$\vert\mathcal{N}\vert$	$\vert\mathcal{A}\vert$	$\vert\mathcal{K}\vert$
R04	10	60	10	R10	20	120	40
R05	10	60	25	R11	20	120	100
R06	10	60	50	R12	20	120	200
R07	10	82	10	R13	20	220	40
R08	10	83	25	R14	20	220	100
R09	10	83	50	R15	20	220	200

Table 3.2 Details of the normal distributions used for generating “timed” instances

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

the transportation cost for a carrier that spends 0.55 cents per mile and their trucks travel at 60 miles per hour. For each commodity $k \in \mathcal{K}$, we followed a normal distribution to generate the available time e^k randomly. 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 $\boldsymbol{\overline{\tau}}$. We set the due time of each commodity $k \in \mathcal{K}$ by $l^k = e^k + \mathscr{L}_k + \mathscr{F}_k$, where $\mathscr{F}_k \geq 0$ represents the corresponding time flexibility, which we set randomly also by a normal distribution. Here, we used the same normal distribution to generate the available times e^k for all instances, but used three different normal distributions to generate $\mathscr{F}_k$ for instances of high, medium and low time flexibility, respectively. As such, we had three combinations of the normal distributions to generate available times and commodity time flexibility. The details of these normal distributions are shown in Table 3.2, where $\mathscr{L}$ represents the average of $\mathscr{L}_k$ over all $k \in \mathcal{K}$.

For each “timed” instances, we then generated the in-storage holding costs and the late arrival penalty costs. To generate the in-storage holding costs, we applied the similar method to the one used in Section 2.4.1. For each $k \in \mathcal{K}$, we set the per-unit-of-demand-and-time cost h_i^k equal to $0.3\,\epsilon_i$ for $i \in \mathcal{N}$ with $\epsilon_i = 1/|\mathcal{A}_i| \sum_{a \in A_i}(c_a + f_a/u_a)/\tau_a$, and set $h_{d^k}^k = 0$. Inspired by Lanza et al. [76], we set the penalty cost g^k per unit of time for the delay of each commodity $k \in \mathcal{K}$ to be proportional to the most expensive transportation cost per-unit-of-time for it passing through a service, i.e., $g^k = \mu_1^k \times max_{a \in \mathcal{A}}\{(c_a * q^k + f_a \lceil q^k/u_a \rceil)/\overline{\tau}_a\}$, where parameter $\mu_1^k = 2$.

To define the uncertainty set of travel time, we needed to generate the deviation of the travel time and the budgeted of uncertainty. For each arc $(i, j) \in \mathcal{A}$, the deviation of the travel time $\hat{\tau}_{ij}$ was set to be $\hat{\tau}_{ij} = \hat{\mu}_{ij}\overline{\tau}_{ij}$, where $\overline{\tau}_{ij}$ is the nominal value of travel time generated, and $\hat{\mu}_{ij}$ is a coefficient randomly picked from 0.1 to 0.5. The budgeted of uncertainty Γ was set to be $\lfloor \mu_2 \cdot |\mathcal{K}| \rfloor$ with $\mu_2 = 0.1$.

Accordingly, for each network in each problem class, we randomly generated 3 instances for each combination of distributions for available times and time flexibility. Thus, we obtained $5 \times 3 \times 3 = 45$ instances for each problem class and $12 \times 45 = 540$ instances in total. These instances were grouped by the scale of the problem class, i.e., the scale of the physical network and the commodity set, resulting in three groups of instances, namely small-sized instances (i.e., those generated based on networks in R4, R5, R7, R8), medium-sized instances (i.e., those generated based on networks in R6, R9, R10, R13) and large-sized instances (i.e., those generated based on networks in R11, R12, R14, R15).

3.7.2 Computational Performance

Computational Results

To evaluate the effectiveness and efficiency of the EC&CG and C&CG methods for the robust CTSNDP-HC, we applied the two solution methods on the newly generated instances, with the threshold for the optimality gap set to be 0.01% and the time limit set to be 4 hours. This time limit was imposed over all the iterations, and at each iteration the time limit imposed on for solving the master problem was 2 hours.

Table 3.3 Computational results on small-sized instances

Class	EC&CG			C&CG		
	Opt	Iter	Time	Opt	Iter	Time
R4	100%	3.2	0.3	100%	3.3	0.5
R5	100%	8.4	217.4	100%	8.4	728.1
R7	100%	4.3	0.7	100%	4.4	1.0
R8	100%	8.8	115.2	100%	8.9	413.7

When the time limit of 4 hours was reached, the algorithm stopped solving the master problem immediately, but it still evaluated the worst-case second-stage cost values for all the first-stage feasible solutions obtained in the final iteration, before its final termination. With imposing such a time limit, in each iteration of the algorithm, the lower bound value LB was updated according to the best known lower bound on the optimal objective of the master problem, which was given by the Gurobi solver at termination. For both EC&CG and C&CG methods, we used $(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$ obtained by the procedure in Section 3.5.4 as the initial first-stage solution.

The computational results of the EC&CG and C&CG methods on small-sized instances are shown in Table 3.3. The percentage of instances solved to within an optimality gap of 0.01% for each group is shown in the columns "Opt" and the average number of the iterations for each solution method is displayed in columns "Iter". Table 3.3 indicates that all small-sized instances can be solved to optimality by both EC&CG and C&CG methods with almost same number of iterations. Thus, we only compare the running time over the small-sized instances, which are shown in columns "Time" in CPU seconds, for each solution method. The results in Table 3.3 show that both the EC&CG and C&CG methods are very efficient in solving the small-sized instances, but the EC&CG method is more efficient as it can solve the small-sized instances with significant less computational time.

The computational results on medium-sized instances are shown in Table 3.4. Similarly, in Table 3.4, columns "Opt" and "Iter" present the percentage of instances solved to within an optimality gap of 0.01% and the average number of the iterations

Table 3.4 Computational results on medium-sized instances

Class	EC&CG				C&CG				UB Improvement	
	Opt	Iter	Gap	AVG Time	Opt	Iter	Gap	AVG Time	AVG	MAX
R6	11%	14.2	5.49%	1262.7	11%	8.5	6.48%	3434.3	1.24%	7.00%
R9	36%	15.3	3.41%	685.4	24%	10.2	4.53%	2792.2	1.57%	10.67%
R10	44%	18.0	2.18%	337.3	36%	14.2	2.47%	649.7	0.46%	3.49%
R13	36%	15.8	3.21%	1040.2	31%	12.3	3.53%	3216.4	0.50%	3.28%

for each instance group, respectively. The average gaps between the lower bound LB_{CCG} obtained by the C&CG method and the upper bound (UB) obtained by each solution method are shown by the columns "Gap". Here, we used LB_{CCG} to compute the gaps for both the EC&CG and C&CG methods, since the LB obtained when the EC&CG method stops may not always be a valid lower bound for the original robust optimization problem. Moreover, columns "AVG Time" show the average computational time in CPU seconds over the instances solved to optimality by the C&CG method (note that all these instances were also solved to optimality by the EC&CG method). Column "UB Improvement" indicates the difference of the upper bound values obtained by these two solution methods over the instances which were not solved to optimality by the C&CG method, where "AVG" and "MAX" indicate the group average and maximum values of this difference, respectively. More specifically, for each instance, this difference was calculated by $\frac{UB_{CCG}-UB_{ECCG}}{UB_{CCG}}$ where UB_{ECCG} and UB_{CCG} represent the upper bound values obtained by the EC&CG and C&CG methods, respectively.

The results in Table 3.4 show that, for the medium-sized instances, the EC&CG and C&CG methods are still efficient in producing solutions of good qualities with an average optimality gap varying from 2.18% to 5.49% and 2.47% to 6.48%, respectively. However, the EC&CG method is more efficient. It can solve more instances to optimality in shorter computational time, and can provide better upper bound solutions for those hard instances with the maximum improvement up to 10.67%.

For large-sized instances, although both the EC&CG and C&CG methods cannot solve any instances to optimality, the comparison between the upper bound solutions

Table 3.5 Computational results on large-sized instances

Class	EC&CG		C&CG		UB Improvement	
	Iter	Failure	Iter	Failure	AVG	MAX
R11	3.1	0%	2.4	0%	5.63%	22.48%
R12	2.2	0%	1.6	22%	5.61%	14.18%
R14	5.2	0%	3.2	0%	6.05%	17.37%
R15	2.6	0%	1.4	33%	7.86%	20.66%

provided by the two solution methods can still show the effectiveness and efficiency of our proposed solution methods. Table 3.5 displays the comparison results on large-sized instances using the same notations introduced in Table 3.4. In addition, columns "Failure" report the percentage of the instances for which the Gorubi solver could not solve the root linear programming relaxation problem of the master problem in the first iteration within the time limit of 4 hours. For these failure cases, the algorithm returned the initial solution $(\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)})$ obtained by the procedure proposed in Section 3.5.4. From Table 3.5, we can see that the C&CG method fails to provide any new feasible solution other than the initial solution for 22% of R12 instances and 33% of R15 instances, while the EC&CG method is able to provide solutions better than the given initial solution for all the considered instances. The average improvements in the upper bound archived by the EC&CG method relative to that obtained by the C&CG method vary from 5.61% to 7.86% for the four instance sets, with a maximum improvement up to 22.48%. These analyses show that the EC&CG method significantly outperforms the C&CG method, in terms of producing solutions of much better qualities for all considered large-sized instances within the time limit.

Moreover, Table 3.6 shows the computational time for solving the master problems and the subproblems $\text{SRP}_1(\boldsymbol{x}, \boldsymbol{z})$, in terms of the percentages over the total computational time, which are presented in columns "MP Time" and "RP Time", respectively. Table 3.6 indicates that for both solution methods, most of the computing time is taken up by solving the master problems. The small and stable proportion of the

Table 3.6 Time of solving the master problems and the subproblems

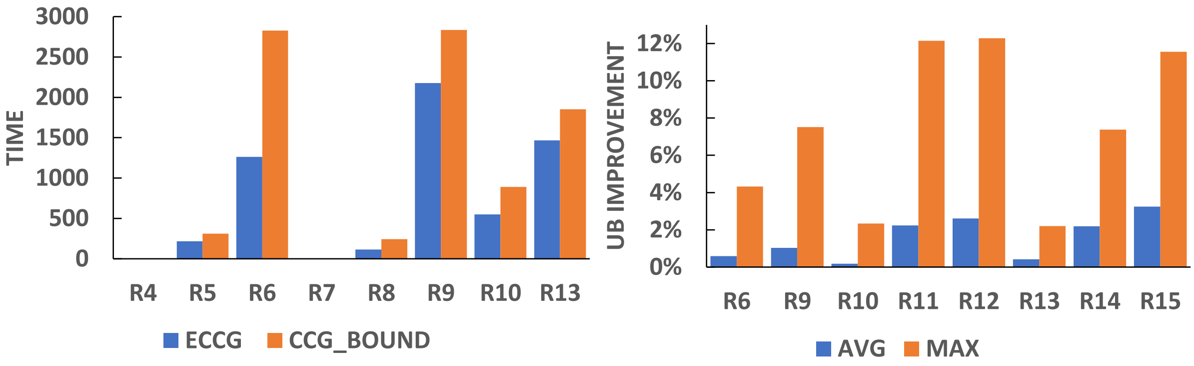
Class	EC&CG		C&CG	
	MP Time	RP Time	MP Time	RP Time
R6	98.57%	1.39%	99.38%	0.59%
R9	96.90%	2.94%	99.03%	0.77%
R10	86.00%	13.45%	90.94%	8.39%
R11	95.36%	3.86%	98.43%	0.73%
R12	91.83%	6.82%	97.91%	0.47%
R13	94.42%	5.32%	97.94%	1.91%
R14	97.96%	2.01%	99.56%	0.36%
R15	93.94%	5.44%	98.82%	0.26%

computational time consumed for solving the subproblems implies the effectiveness of the size reduction strategy proposed in Section 3.6.1.

Overall, the results of our computational experiments demonstrate the effectiveness and efficiency of the EC&CG and C&CG methods. In comparison of both computational time and solution quality, the EC&CG method outperforms the standard C&CG method. As shown in columns "Iter", this outstanding performance of the EC&CG method comes from that the EC&CG method can execute more iterations within the time limit, thus is more likely to find better upper bound solutions.

Performance of Dynamic Parameter Adjustment

The EC&CG method enhances the C&CG method by introducing the parameter $\mathbf{\Pi}$, named as consolidation frequency bound. The enhanced effectiveness and efficiency of the EC&CG method are due to both the upper bound derived for $\mathbf{\Pi}$ and the dynamic parameter adjustment process developed for $\mathbf{\Pi}$. The upper bound $\mathbf{\Pi}^*$ derived for $\mathbf{\Pi}$ can directly reduce the solution space and hence accelerate the algorithm. To examine the impact of the dynamic parameter adjustment process, we next compare the performance of the EC&CG method (with the dynamic parameter adjustment process) with that of the EC&CG method without dynamic parameter adjustment process, which we denote as EC&CG_BOUND.

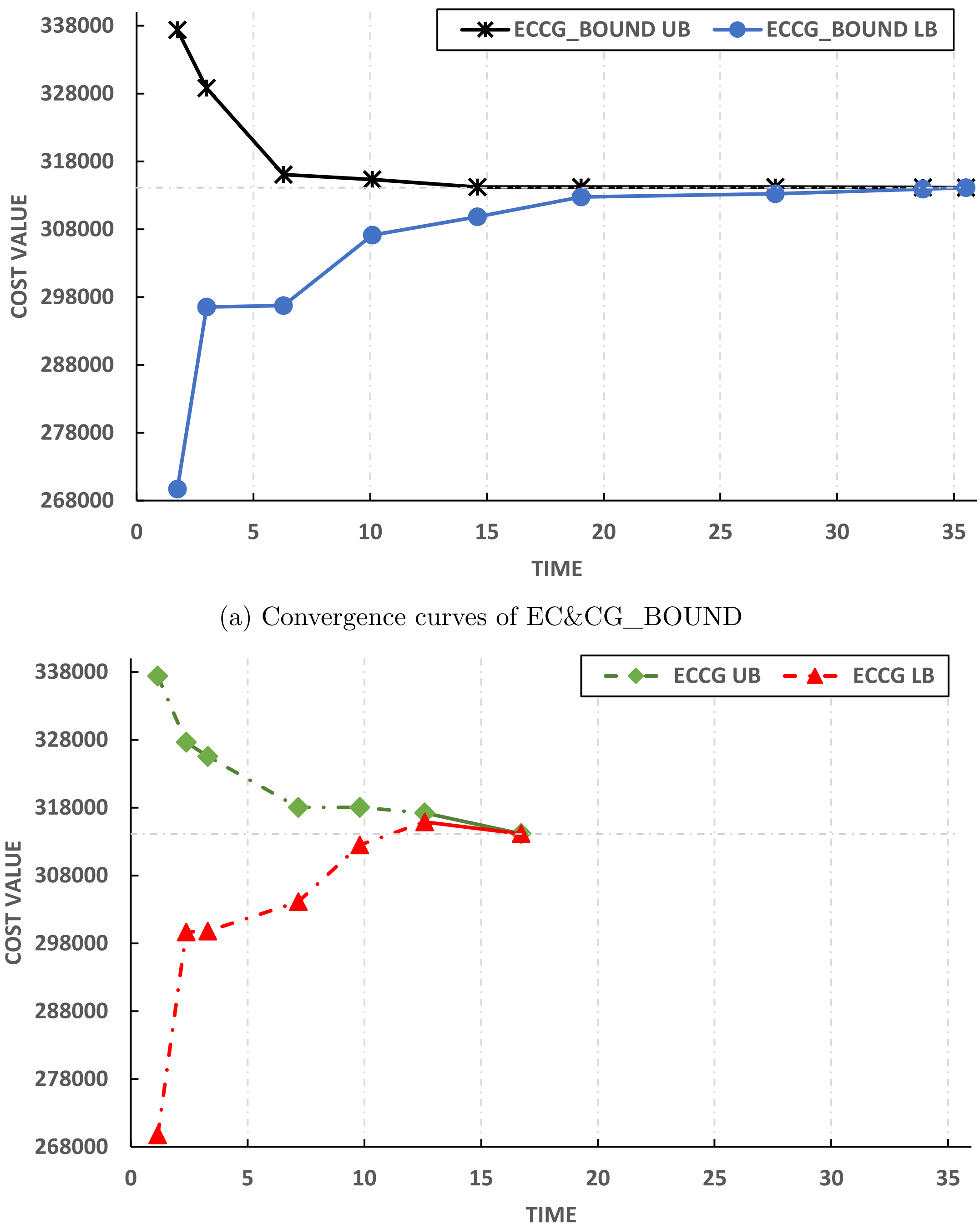


(a) AVG Time (b) UB Improvement

Figure 3.2 Performance comparisons of EC&CG to EC&CG_BOUND

Figure 3.2(a) and (b) demonstrate the corresponding improvements in computational time and solution quality due to applying the dynamic parameter adjustment process. Figure 3.2(a) indicates the average computational time in CPU seconds over the instances solved to optimality by the EC&CG_BOUND method. All these considered instances were also solved to optimality by the EC&CG method. For R4 and R7 instances, both solution methods solved all the instances with an average computational time of less than 1 second, thus the corresponding bar charts for R4 and R7 instances are not apparent. Figure 3.2(b) indicates the improvement in the upper bound achieved by the EC&CG method over the instances which were not solved to optimality by the EC&CG_BOUND method, where "AVG" and "MAX" present the group average and maximum value of this improvement, respectively, for the corresponding instance set. While Figure 3.2(a) demonstrates a significant saving in computational time, Figure 3.2(b) shows a significant improvement in upper bound for the considered instances with a maximum improvement up to 12.28%. Therefore, Figure 3.2 reveals the significant impact of the dynamic parameter adjustment process in accelerating the solution methods and in producing better upper bound solutions.

Furthermore, Figure 3.3 illustrates how the dynamic parameter adjustment affects the convergence behavior of the solution method, by comparing the EC&CG_BOUND method (without dynamic parameter adjustment) and the EC&CG method (with dynamic parameter adjustment). It plots LB and UB curves for each of the two solution



(a) Convergence curves of EC&CG_BOUND

(b) Convergence curves of EC&CG

Figure 3.3 Convergence curves over time

methods based on the computational results over a selected problem instance. In both Figures 3.3(a) and 3.3(a), each data point on the curves indicates a pair of a value (LB or UB) obtained in an iteration and the time that the iteration completes. The dotted curves of the EC&CG method in Figure 3.3(b) correspond to the period that processes the dynamic adjustment of the parameter $\mathbf{\Pi}$, whereas the solid curves there correspond to the period when parameter $\mathbf{\Pi}$ has been adjusted to the maximum consolidation frequency $\mathbf{\Pi}^*$, so that the algorithm follows the standard C&CG method.

From Figure 3.3, we have the following observations, showing the impact of the dynamic parameter adjustment process:

1. By comparing the average time spent on each iteration of the two methods, we can see that the dynamic parameter adjustment process shortens the time spent by the EC&CG method in each iteration. This is because the master problems solved in the EC&CG method have fewer variables and constraints than those in the EC&CG_BOUND method.

2. As shown the dotted curves in Figure 3.3(b), during the process of the parameter adjustment, the lower bound obtained by the EC&CG method may be larger than the optimal objective value of the original robust optimization problem, which is due to the restriction of the solution space. However, the lower bound value is later corrected after the parameter $\mathbf{\Pi}$ is adjusted to be the maximum consolidation frequency $\mathbf{\Pi}^*$, and the dynamic parameter adjustment process shortens the total running time of the EC&CG method significantly.

3. We know that the EC&CG method searches the feasible solutions within a smaller solution space in each iteration but postpones verification of the optimality of the solution obtained. Due to this, one may expect that the EC&CG method requires more iterations than the EC&CG_BOUND method to obtain the optimal solution and prove its optimality. However, the example shown in Figure 3.3 indicates that, for some problem instances, the EC& CG method can obtain optimal solution with even fewer iterations than the EC&CG_BOUND method.

This reveals that, in some situations, the dynamic adjustment process of the parameter $\mathbf{\Pi}$ can help the algorithm to find effective worst-case scenarios as well and cutting planes earlier, thereby speeding up the convergence of the algorithm.

3.7.3 The Price of Robustness

The goal of the robust optimization for the CTSNDP-HC under travel time uncertainty is to optimize the solution performance under the worst-case scenario, which may sometimes be too conservative and at a price of poor performance under scenarios other than the worst-case one. Following Bertsimas and Sim [16] and Atamtürk and Zhang [5], in this section, we evaluate the performance of the first-stage robust solution obtained by our proposed model RO under different scenarios to assess the advantage and the price of its robustness.

Performance on Nominal Scenario

We first compare the performance of the robust solution obtained by model RO with that of the solution obtained by the deterministic model DO under both the worst-case scenario and the nominal scenario. We define two performance indicators, UBD and ND, as follows:

$$\begin{aligned} UBD &= \frac{F_Z(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}) - F_Z(\boldsymbol{x}, \boldsymbol{z})}{\mathcal{Z}(\boldsymbol{x}, \boldsymbol{z})}, \text{ and} \\ ND &= \frac{\mathcal{Z}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}) - \mathcal{Z}(\boldsymbol{x}, \boldsymbol{z})}{\mathcal{Z}(\boldsymbol{x}, \boldsymbol{z})} \end{aligned}$$

where $(\boldsymbol{x}, \boldsymbol{z})$ is the optimal solution obtained by model DO with a total cost of $\mathcal{Z}(\boldsymbol{x}, \boldsymbol{z})$ under the nominal scenario, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ is the solution obtained by the EC&CG method with a total cost of $\mathcal{Z}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ under the nominal scenario, and $F_Z(\boldsymbol{x}, \boldsymbol{z})$ and $F_Z(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ are the total costs under the worst-case scenario in the budgeted uncertainty set for the corresponding solutions. It can be seen that the value of UBD indicates the cost saving achieved by the optimal robust solution under the worst-case scenario, and that

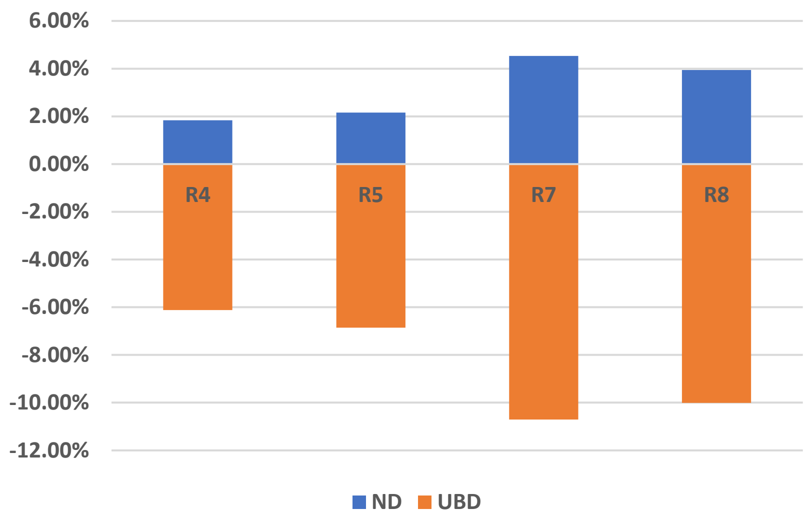


Figure 3.4 Performance of the robust solution under the worst-case scenario and nominal scenario

the value of ND indicates the price of its robustness under the nominal scenario, in comparison against the optimal nominal solution.

Figure 3.4 plots the average values of UBD and ND in percentage over the four classes of the small-sized instances. The values of UBD in Figure 3.4 indicate that the solution provided by model RO can help the decision-maker achieve a considerable amount of total cost reduction in the worst-case scenario. This shows the benefits for incorporating travel time uncertainty in solving the CTSNDP-HC, and demonstrates the robustness of the solution provided by model RO against travel time uncertainty. The values of ND in Figure 3.4 reveal that the robustness of the solution provided by model RO, i.e., its significant amount of cost saving in the worst-case scenario, is at the expense of certain cost increase in the nominal scenario. However, such cost increase in the nominal scenario is significantly less than the cost saving achieved by model RO in the worst-case scenario.

Performance on Random Scenarios

To evaluate the performance of the robust solution under random scenarios, we compare the solutions obtained from our proposed model RO and the solutions obtained from a stochastic programming model (which minimizes the expected cost over a set of scenarios). Let $\tilde{\Omega}$ indicate a scenario set consisting of all possible travel time realizations. Let $Prob(\tilde{\boldsymbol{\tau}})$ denote the probability of each possible realization $\tilde{\boldsymbol{\tau}} \in \tilde{\Omega}$. The stochastic programming model for the CTSNDP-HC under travel time uncertainty can be formulated as below:

[SP]

$$\min \sum_{\tilde{\boldsymbol{\tau}} \in \tilde{\Omega}} Prob(\tilde{\boldsymbol{\tau}}) \Big(\sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} (c_{ij}^k q^k) x_{ij}^k + \sum_{(i,j) \in \mathcal{A}} \sum_{r=1}^{|\mathcal{K}|} f_{ij} y_{ijr} + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} (h_i^k q^k) w_i^{k(\tilde{\tau})} + \sum_{k \in \mathcal{K}} g^k s^{k(\tilde{\tau})} \Big)$$

$$s.t. \; (\boldsymbol{v}^{(\tilde{\tau})}, \boldsymbol{b}^{(\tilde{\tau})}, \boldsymbol{w}^{(\tilde{\tau})}, \boldsymbol{s}^{(\tilde{\tau})}) \in \mathcal{Q}(\tilde{\boldsymbol{\tau}}), \quad \forall \; \tilde{\boldsymbol{\tau}} \in \tilde{\Omega},$$

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

Our evaluation is based on instances in R4, which were all solved to optimality for both model RO and model SP. For each considered instances, we randomly generated 200 scenarios, referred to as the *SP scenarios*. For each scenario $\tilde{\boldsymbol{\tau}}$, each travel time $\tilde{\tau}_{ijr}$, $\forall (i,j) \in \mathcal{A}$ and $r \in \{1, 2, ..., |\mathcal{K}|\}$, was drawn uniformly from $[\overline{\tau}_{ij} - \hat{\tau}_{ij}, \overline{\tau}_{ij} + \hat{\tau}_{ij}]$. We set $Prob(\tilde{\boldsymbol{\tau}}) = \frac{1}{200}$, so that every scenario was with equal probability. By the same approach, we also randomly generated 1000 *testing scenarios*.

We used the Gurobi solver to solve model SP over the 200 SP scenarios, and applied our EC&CG method to solve model RO for each uncertainty budget $\Gamma \in \{1, 2, ..., 10\}$. We compare the solutions obtained by the two models in threefold: (1) the average performance over the all the testing scenarios; (2) the worst performance over all the testing scenarios; (3) the performance under the worst-case scenario in the given budgeted uncertainty set. For every solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ obtained from model RO or model SP, we computed its cost value under each testing scenario by solving the deterministic model, DO, with $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ fixed as the given solution, with the travel times equal to the realized travel times in the corresponding scenario, and with the variables and constraints related to the delay penalty incorporated. We also computed the worst-case second-stage cost of the given solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ by solving model $\text{SRP}_1(\boldsymbol{x}, \boldsymbol{z})$, which

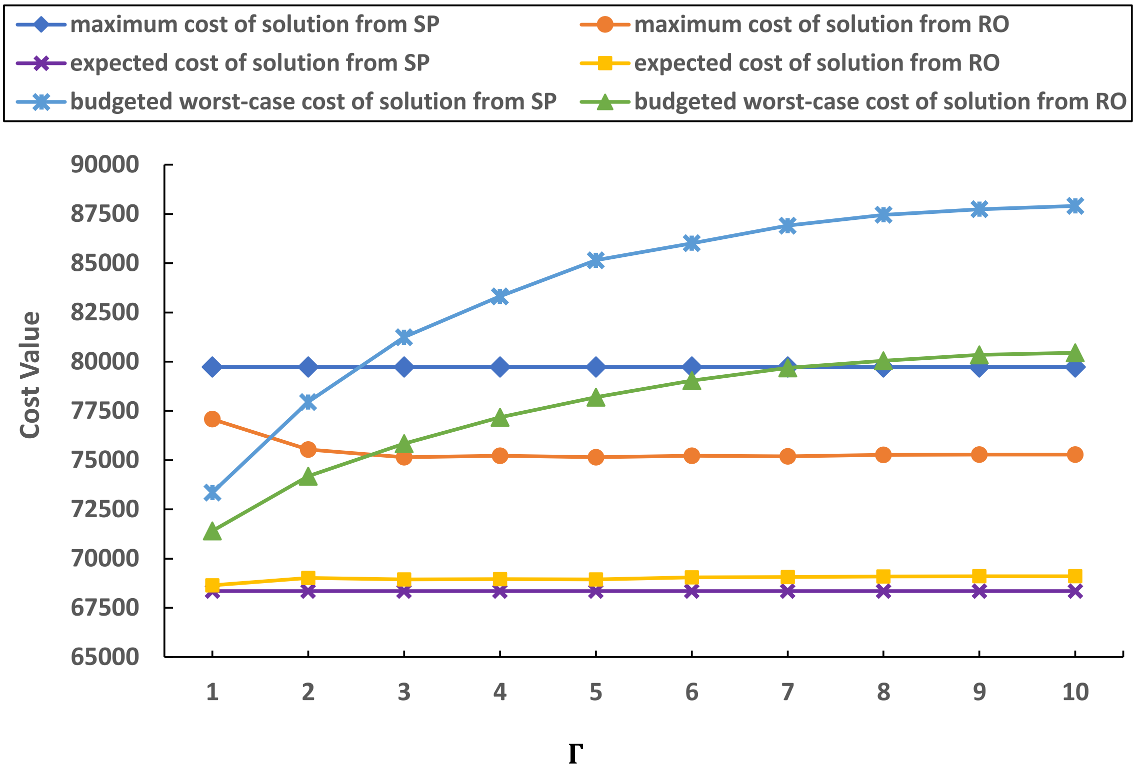


Figure 3.5 Comparison of robust solutions and stochastic solutions

was then used to compute the total cost of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ under the worst-case scenario in the given budgeted uncertainty set.

Figure 3.5 plots the expected total cost and the maximum total cost over the 1000 testing scenarios, as well as the total cost under the worst-case scenario in the given budgeted uncertainty set $\mathbb{U}(\Gamma)$, for the solutions obtained from model RO and model SP, with values averaged over all instances in group R4. Not surprisingly, compared with the solutions obtained from model RO, the solutions obtained from model SP have poorer worst-case performance over the given budgeted uncertainty set $\mathbb{U}(\Gamma)$. This further verifies the robustness of the solution provided by model RO over the budgeted uncertainty set $\mathbb{U}(\Gamma)$. For the random testing scenarios, the expected objective values of the robust solutions obtained from model RO under $\Gamma \in \{1, 2, ..., 10\}$ are larger but very close to that of the solutions obtained from model SP. Thus, we can conclude that the cost saving achieved by the robust optimization for the worst-case scenario is only at the expense of a slightly worse expected performance over such a random scenario set. Moreover, the solutions obtained from model SP also have a larger worst-case total cost over the random scenario set. The robust average performance of the solution obtained from model SP is thus associated with a significant risk of a poor worst-case performance over a given scenario set or budgeted uncertainty set. Compared with the solution obtained from model SP, the robust solution obtained from model RO achieves a better trade-off between robustness and the price of robustness.

Moreover, we observe that, with a small value of Γ, i.e., $\Gamma = 1$ or 2, the total cost of the solution provided by either of the two models under the worst-case scenario in the given budgeted uncertainty set $\mathbb{U}(\Gamma)$ is lower than the maximum (or worst-case) total cost among the testing scenarios, while the opposite occurs for a larger value of Γ, i.e., $\Gamma \geq 5$. This implies that when Γ is too small, the uncertainty set $\mathbb{U}(\Gamma)$ cannot cover sufficient scenarios to achieve robustness, and that when Γ is too large, the corresponding robust optimization models becomes too conservative, leading to solutions with over-conservatism. Thus, the decision-maker have to choose Γ properly to incorporate uncertainty sufficiently as well as to avoid having a robust optimization model that is too conservative. However, the optimal value of Γ currently can only be determined by numerical experiments. Finally, it is worth noting that the solution quality of model SP relies heavily on the accuracy of the probability distribution and the completeness of the considered scenario set. From this perspective, model RO also has an advantage, as it requires much less information on distribution and is easier to be solved for much larger-sized instances.

3.8 Summary

This chapter studied on how to handle travel time uncertainty for the continuous-time service network design problem with holding costs incorporated (CTSNDP-HC). In real-life applications, uncertain factors, such as weather and traffic conditions, often result in fluctuated travel times and thus cause delays of transportation services. Handling travel time uncertainty is an important but challenging task in the design of robust and cost-effective service network. Despite its importance, the robust CTSNDP-HC under travel time uncertainty has not been studied in the literature. This may be due to its modeling complexity as well as to the ineffectiveness of the time-expanded network in handling travel time uncertainty.

To tackle this challenge, instead of using the TI formulation based on the time-expanded network, we newly proposed a consolidation-index formulation for the deterministic CTSNDP-HC in this study. The new formulation is based on the flat network, and it models the temporal component of the problem by a set of consolidation-index variables and constraints. Based on this new deterministic formulation, we derived a two-stage MINLP formulation for robust CTSNDP-HC with all the possible travel time realizations described by a probability-free budgeted uncertainty set. The first stage of this robust optimization model optimizes the selection of services, as well as the routing and consolidation plans of commodities. The second stage determines the departure schedule after the actual values of the travel times are realized. This two-stage MINLP formulation for the robust CTSNDP-HC was further linearized and solved by a standard column-and-constraint generation method. We then introduced a parameter $\mathbf{\Pi}$, referred to as the consolidation frequency bound, into the two-stage robust optimization model. With this parameter, we enhanced the C&CG method by dynamically adjusting $\mathbf{\Pi}$. An extensive computation study was conducted to evaluate the performance of the newly proposed two-stage robust optimization model and solution methods. The experimental results showed that the enhanced C&CG method outperforms the standard C&CG method significantly in terms of shorter computational time and better solution quality. Moreover, the performance of the robust solution under different scenarios demonstrated that the value for considering travel time uncertainty for the CTSNDP-HC is more significant than the price of the robustness against travel time uncertainty.

Chapter 4

Conclusions and Future Works

In this chapter, we summarize the major results and findings of this dissertation and discuss some future research directions.

4.1 Major Results and Findings

In this dissertation, we conducted two studies to develop exact solution methods for the deterministic CTSNDP with holding costs incorporated, and for the robust CTSNDP-HC under travel time uncertainty, respectively.

The deterministic CTSNDP aims to optimize the establishment of transportation services as well as the distribution and consolidation of commodity flows for a carrier, so that the total operational cost over a continuous-time planning horizon is minimized. Such a carrier often relies on shipment consolidation to maintain its profitability. This leads to the waiting around of shipments, which incurs holding costs. For many practical applications of the CTSNDP, holding costs are a vital part of the total cost and significantly affect the decisions on opening services, as well as on the transportation and consolidation of shipments. Despite their importance, holding costs has not been taken into account in existing studies on the CTNSDP, as introducing the holding costs significantly complicates the problem and makes it very challenging to solve. In the first study of this dissertation, we tackled the challenge of investigating how to develop an efficient exact algorithm for the CTSNDP with holding costs incorporated.

Unlike the classic SNDP, which is often defined on a planning horizon that is discretized into a finite number of equal-length time intervals, the CTSNDP is defined on a continuous-time planning horizon so as to eliminate approximation errors caused by the discretization. Existing algorithms developed for the CTSNDP and its variants mainly follow a dynamic discretization discovery (DDD) solution framework. This DDD framework iteratively refines a finite set of integral time units for discretization, constructs a partially time-expanded network based on the discretization, and uses the network to derive relaxations and feasible solutions to the problem. Such DDD algorithms are known to be valid only under the assumption that holding costs are zero, which significantly restricts the applications of these algorithms in practice. We have shown that, with the holding costs ignored, there can be a significant loss as a result of the total cost of actual operation of the service network design. However, incorporating freight holding costs in solving the CTSNDP is challenging. With freight holding costs incorporated, the existence of the finite complete time-index model for the problem is not clear to see, and the relaxation and refinement methods in the DDD algorithms for the CTSNDP also become invalid.

To tackle this challenge, in the first study of this dissertation, we first modeled the CTSNDP-HC by a time-index formulation based on a discretized time-expanded network. By utilizing the total unimodularity of a linear programming model, we proved that to obtain an exact optimal solution to the CTSNDP-HC, it is sufficient to include only integral time units in the time-index formulation, which is referred to as the fully discretized time-expanded network. Based on a partially time-expanded network consisting of only a subset of time points of the fully discretized time-expanded network, we then derived a relaxation of the CTSNDP-HC, which provides a lower bound on the total cost of the optimal solution to the CTSNPD-HC. With these, together with a new upper bound heuristic and a new discretization refinement procedure that take the holding costs into account, we developed a new DDD algorithm that can solve the

CTSNDP-HC to optimality. The effectiveness and efficiency of the proposed DDD algorithm in both finding optimal solutions to the CTSNDP-HC and producing tight lower and upper bounds were validated through extensive computational experiments. These computational results also showed the significant impact of holding costs on the decisions of routing and consolidation. The results also indicated that ignoring the holding costs in the CTSNDP will lead to poor quality solutions, especially for those instances with a concentrated transit network and a long delivery due time. This work not only enriches the optimization techniques for the deterministic CTSNDP-HC, but also enhances the DDD algorithm and extends its practical applications.

While the first study of this dissertation explored the CTSNDP-HC with deterministic parameters, the second study of this dissertation investigated the robust solution for the CTSNDP-HC under travel time uncertainty. Due to various uncertain factors, such as weather and traffic conditions, travel time varies considerably, and is one of the prime sources of uncertainty in service network design. These uncertain travel times often cause delays to transportation services in actual operation, so that commodities may not be delivered on time. Such disruptive impacts can result in order cancellations or outsourcing, which are both very costly. Despite its importance and its disruptive impact, due to its complexity, travel time uncertainty has seldom been taken into account in the literature on SNDP. There is no existing robust optimization method for the continuous-time SNDP under travel time uncertainty. To complete this research gap, in the second study of this dissertation, we developed a robust optimization model and its solution algorithms for the CTSNDP-HC with uncertain travel times.

The time-index formulations of the CTSNDP-HC, including those known in the existing literature and the one proposed in the first study of this dissertation, cannot be directly utilized to develop robust optimization models and their solution methods for the same problem under travel time uncertainty. To tackle this challenge, in the second study of this dissertation we first introduced a new MILP model for the deterministic CTSNDP-HC, based on the consolidation index in the flat network rather than the time index. From this new deterministic formulation, we were able to develop a two-stage robust optimization model for the robust CTSNDP-HC, with uncertain travel times incorporated in a budgeted uncertainty set. The first stage of the model determines the selection of services as well as the routing and consolidation plans for commodities. The second stage of the model determines the departure schedule of each selected service after the actual travel times are realized. To solve this two-stage robust optimization model efficiently, we first developed a column-and-constraint generation method (C&CG), from which we then proposed an enhanced C&CG method by parameterizing the robust optimization model and utilizing some novel optimization techniques for dynamic parameter adjustment. Results from the computational study demonstrated the robustness of the solutions obtained from the proposed two-stage robust optimization model against travel time uncertainty, and exhibited the efficiency and effectiveness of the two proposed solution methods. The computational results showed that the performance of the enhanced C&CG method dominates that of the C&CG method in terms of shorter computational time and better solution quality. Further analysis of the results revealed that by applying the dynamic parameter adjustment process, the performance of the enhanced C&CG method can be significantly improved, and that the price of such robustness against travel time uncertainty is significantly less than its benefits.

The second study of this dissertation presents the first adaptive robust optimization method for the SNDP with uncertain travel times and non-zero holding costs over a continuous-time planning horizon. The consolidation-index formulation as well as the dynamic parameter adjustment scheme introduced in this work are new to the literature, providing a solid foundation for applying robust optimization techniques to solving various transportation network design problems under travel time uncertainty.

4.2 Future Research Directions

The works presented in this dissertation have opened up three main directions for further research, as explained below.

To Improve the Performance of the Proposed Solution Methods

First, in future research we could focus on improving the performance of the proposed solution methods. For the deterministic CTSNDP-HC, our computational study has demonstrated that the current implementation of the DDD algorithm for the CTSNDP-HC is quite effective. However, some enhancements could be conducted to further improve its efficiency, such as: 1) We could consider implementing a two-stage DDD algorithm. The first stage would utilize the DDD algorithm proposed by Boland et al. [20] to solve the CTSNDP with the holding costs set to be zero. Then the second stage would apply our proposed DDD algorithm in solving the CTSNDP-HC by taking the solution and final network of the first stage as the initial setting. The size of the final MIP might be significantly reduced by this two-stage scheme, thereby improving the performance of the algorithm; 2) We proposed only a default refinement strategy in the first study of this dissertation. The algorithm can be further accelerated by utilizing a more careful and clever refinement strategy which can obtain the optimal solution by adding fewer time points or conducting fewer iterations; 3) The proposed DDD algorithm never removes time points from the partially time-expanded network after they have been added. The size of the partially time-expanded network in each iteration will affect the size of the resulting MIP model and thus the computational efficiency of the algorithm. This implies that the algorithm framework can be further enhanced in the future so that some superfluous time points in the current partially time-expanded network can be removed in the next iteration to reduce the sizes of the MIPs.

Likewise, for the robust CTSNDP-HC, it would be of interest to further improve the performance of the proposed EC&CG method, for which several recommended directions are mentioned here: As shown in Section 3.7.2 of the second study of this dissertation, the EC&CG method's bottleneck lies in solving the master problem, because the big-M are used in some of its constraints, and the number of variables and constraints increases quickly over iterations. To tackle this, one possible approach would be to design more efficient methods, such as branch-and-price algorithms or some heuristics, to solve the master problem, instead of solving it directly by the MIP solver. Another possible approach is to investigate whether any worst-case scenarios, together with their corresponding variables and constraints, can be removed from the master problems in the later iterations, so as to reduce the sizes of the master problems. For this, it would be worth developing effective domination rules among the worst-case scenarios. Moreover, another promising future research direction would be to develop more efficient strategies for the parameter adjustment, which could significantly affect the performance of the EC&CG method.

To Develop New Optimization Techniques and Models

Another natural direction for future research would be to develop other new optimization techniques for the deterministic CTSNDP-HC and for its robust variation under travel time uncertainty. The DDD algorithm for the deterministic CTSNDP proposed by Boland et al. [20] has recently been further modified by Marshall et al. [81], where the time-expanded networks are defined on time intervals instead of time points. It is also possible to adapt our newly proposed DDD algorithm for the deterministic CTSNDP-HC to utilize the interval-based time-expanded network. As part of our future work, we will investigate such an interval-based DDD algorithm for the deterministic CTSNDP-HC, comparing its performance against the algorithm proposed in this dissertation, and exploring their relationships.

For the robust CTSNDP-HC studied in this dissertation, we optimized the solutions for the worst-case scenario in a given budgeted uncertainty set. To provide the decision-maker with more robust options, it is an attractive research direction to investigate a distributionally robust optimization method for tackling the CTSNDP-HC under travel time uncertainty, so that solutions are optimized for the worst-case probability distribution within the family of distributions. It is widely recognized that solutions obtained from the distributionally robust optimization method are likely to be less conservative than those obtained from robust optimization [53, 102]. Besides, affine decision rules [9], which impose an affine dependence of the second-stage decisions on the predefined uncertainties, provide an alternative method to formulate and solve the robust problems. The optimality of affine decision rules in well-defined two-stage robust optimization has been extensively investigated in previous studies [13, 93, 55]. If we can prove the existence of an optimal affine solution to the second-stage of the robust CTSNDP-HC, applying affine decision rules may help us to develop more efficient solution methods to solve the robust CTSNDP-HC. Moreover, Long et al. [80] have recently proposed a robust satisficing framework which aims to minimize the fragility of the robust model to uncertainty in achieving the prescribed target. It is of great interest for us to apply this new framework in solving the CTSNDP-HC under travel time uncertainty. For these promising future research directions, we believe that our newly derived consolidation-index deterministic model and newly proposed dynamic parameter adjustment scheme can provide a solid foundation.

In addition, for both the deterministic CTSNDP-HC and robust CTSNDP-HC, to make the proposed optimization models more useful for practical applications, it is of great interest to develop more efficient solution methods to solve much larger instances. Sarayloo et al. [91] have recently proposed a learning-based matheuristic for the multi-commodity network design problem with stochastic demands. They proved that this method is highly effective in finding good-quality solutions, especially for large instances. As a future work, we will explore the development of an effective matheuristic for both the deterministic CTSNDP-HC and the robust CTSNDP-HC.

To Investigate More Applications

In our future studies, we will investigate different variants of the CTSNDP and other optimization problems based on the methods and results developed in this dissertation. Real-world service network design problems pose several challenging extensions, such as asset management, terminal capacities, and compatibility of commodities. Our future work can therefore go in the direction of considering these extensions, as well as other important features of these problems. In particularly, the CTSNDP with time-dependent travel times, i.e., the travel time functions are continuous functions of departure times or link flows, can be more challenging and is of interest for future research. In this dissertation, we only consider travel time uncertainty within a budgeted uncertainty set. However, real-world service network design problems encounter more uncertainties than just those of travel times. Thus, there would be significant research value and practical value in incorporating other uncertainties, such as demand uncertainty, into the proposed robust optimization model, so as to make it more comprehensive. Besides, it also looks promising to investigate more applications of the relaxation method derived for the deterministic CTSNDP-HC based on the time-expanded network, as well as more applications of the consolidation-index formulation newly proposed for handling travel time uncertainty. These newly proposed modeling techniques can be adopted to other routing and scheduling problems, i.e., liner service network design problem, to handle holding costs and travel time uncertainty.

References

[1] Adulyasak, Y. and Jaillet, P. (2016), Models and algorithms for stochastic and robust vehicle routing with deadlines, *Transportation Science* **50**(2), 608–626.

[2] Andersen, J., Christiansen, M., Crainic, T. G. and Grønhaug, R. (2011), Branch and price for service network design with asset management constraints, *Transportation Science* **45**(1), 33–49.

[3] Andersen, J., Crainic, T. G. and Christiansen, M. (2009*a*), Service network design with asset management: Formulations and comparative analyses, *Transportation Research Part C: Emerging Technologies* **17**(2), 197–207.

[4] Andersen, J., Crainic, T. G. and Christiansen, M. (2009*b*), Service network design with management and coordination of multiple fleets, *European Journal of Operational Research* **193**(2), 377–389.

[5] Atamtürk, A. and Zhang, M. (2007), Two-stage robust network flow and design under demand uncertainty, *Operations Research* **55**(4), 662–673.

[6] Bai, R., Wallace, S. W., Li, J. and Chong, A. Y.-L. (2014), Stochastic service network design with rerouting, *Transportation Research Part B: Methodological* **60**, 50–65.

[7] Belieres, S. (2019), Mathematical programming for tactical transportation planning in a multi-productsupply chain, PhD thesis, Automatic Control Engineering. INSA de Toulouse.

[8] Ben-Tal, A., El Ghaoui, L. and Nemirovski, A. (2009), *Robust optimization*, Princeton university press.

[9] Ben-Tal, A., Goryashko, A., Guslitzer, E. and Nemirovski, A. (2004), Adjustable robust solutions of uncertain linear programs, *Mathematical programming* **99**(2), 351–376.

[10] Ben-Tal, A. and Nemirovski, A. (1998), Robust convex optimization, *Mathematics of operations research* **23**(4), 769–805.

[11] Ben-Tal, A. and Nemirovski, A. (1999), Robust solutions of uncertain linear programs, *Operations research letters* **25**(1), 1–13.

[12] Bertsimas, D. and Brown, D. B. (2009), Constructing uncertainty sets for robust linear optimization, *Operations research* **57**(6), 1483–1495.

[13] Bertsimas, D. and Goyal, V. (2012), On the power and limitations of affine policies in two-stage adaptive optimization, *Mathematical programming* **134**(2), 491–531.

[14] Bertsimas, D., Iancu, D. A. and Parrilo, P. A. (2010), Optimality of affine policies in multistage robust optimization, *Mathematics of Operations Research* **35**(2), 363–394.

[15] Bertsimas, D. and Sim, M. (2003), Robust discrete optimization and network flows, *Mathematical programming* **98**(1), 49–71.

[16] Bertsimas, D. and Sim, M. (2004*a*), The price of robustness, *Operations research* **52**(1), 35–53.

[17] Bertsimas, D. and Sim, M. (2004*b*), Robust discrete optimization under ellipsoidal uncertainty sets, *Technical Report, MIT* .

[18] Bertsimas, D. and Weismantel, R. (2005), *Optimization over integers.*, Athena Scientific.

[19] Birge, J. R. and Louveaux, F. (2011), *Introduction to stochastic programming*, Springer Science & Business Media.

[20] Boland, N., Hewitt, M., Marshall, L. and Savelsbergh, M. (2017), The continuous-time service network design problem, *Operations Research* **65**(5), 1303–1321.

[21] Boland, N., Hewitt, M., Marshall, L. and Savelsbergh, M. (2018), The price of discretizing time: a study in service network design, *EURO Journal on Transportation and Logistics* pp. 1–22.

[22] Boland, N., Hewitt, M., Marshall, L. and Savelsbergh, M. (2019), The price of discretizing time: a study in service network design, *EURO Journal on Transportation and Logistics* **8**(2), 195–216.

[23] Boland, N., Kalinowski, T. and Kaur, S. (2015), Scheduling network maintenance jobs with release dates and deadlines to maximize total flow over time: Bounds and solution strategies, *Computers & Operations Research* **64**, 113–129.

[24] Boland, N. L. and Savelsbergh, M. W. (2019), Perspectives on integer programming for time-dependent models, *TOP* pp. 1–27.

[25] Bookbinder, J. H. and Higginson, J. K. (2002), Probabilistic modeling of freight consolidation by private carriage, *Transportation Research Part E: Logistics and Transportation Review* **38**(5), 305–318.

[26] Calafiore, G. C. and El Ghaoui, L. (2006), On distributionally robust chance-constrained linear programs, *Journal of Optimization Theory and Applications* **130**(1), 1–22.

[27] Chassein, A. and Goerigk, M. (2016), On the recoverable robust traveling salesman problem, *Optimization Letters* **10**(7), 1479–1492.

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

[29] Chen, X., Sim, M. and Sun, P. (2007), A robust optimization perspective on stochastic programming, *Operations research* **55**(6), 1058–1071.

[30] Chen, X., Sim, M., Sun, P. and Zhang, J. (2008), A linear decision-based approximation approach to stochastic programming, *Operations Research* **56**(2), 344–357.

[31] Chinneck, J. W. (1997), Finding a useful subset of constraints for analysis in an infeasible linear program, *INFORMS Journal on Computing* **9**(2), 164–174.

[32] Chinneck, J. W. and Dravnieks, E. W. (1991), Locating minimal infeasible constraint sets in linear programs, *ORSA Journal on Computing* **3**(2), 157–168.

[33] Crainic, T. G. (2000), Service network design in freight transportation, *European journal of operational research* **122**(2), 272–288.

[34] Crainic, T. G., Frangioni, A. and Gendron, B. (2001), Bundle-based relaxation methods for multicommodity capacitated fixed charge network design, *Discrete Applied Mathematics* **112**(1-3), 73–99.

[35] Crainic, T. G., Fu, X., Gendreau, M., Rei, W. and Wallace, S. W. (2011), Progressive hedging-based metaheuristics for stochastic network design, *Networks* **58**(2), 114–124.

[36] Crainic, T. G., Gendron, B. and Hernu, G. (2004), A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design, *Journal of Heuristics* **10**(5), 525–545.

[37] Crainic, T. G., Hewitt, M., Toulouse, M. and Vu, D. M. (2014), Service network design with resource constraints, *Transportation Science* **50**(4), 1380–1393.

[38] Crainic, T. G. and 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**(3), 225–242.

[39] Dantzig, G. B. (1955), Linear programming under uncertainty, *Management science* **1**(3-4), 197–206.

[40] Dash, S., Günlük, O., Lodi, A. and Tramontani, A. (2012), A time bucket formulation for the traveling salesman problem with time windows, *INFORMS Journal on Computing* **24**(1), 132–147.

[41] Delage, E. and Ye, Y. (2010), Distributionally robust optimization under moment uncertainty with application to data-driven problems, *Operations research* **58**(3), 595–612.

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

[43] Dyer, M. and Stougie, L. (2006), Computational complexity of stochastic programming problems, *mathematical programming* **106**(3), 423–432.

[44] Erera, A., Hewitt, M., Savelsbergh, M. and Zhang, Y. (2013), Improved load plan design through integer programming based local search, *Transportation Science* **47**(3), 412–427.

[45] Escudero, A., MuñUzuri, J., Guadix, J. and Arango, C. (2013), Dynamic approach to solve the daily drayage problem with transit time uncertainty, *Computers in industry* **64**(2), 165–175.

[46] Eshtehadi, R., Fathian, M. and Demir, E. (2017), Robust solutions to the pollution-routing problem with demand and travel time uncertainty, *Transportation Research Part D: Transport and Environment* **51**, 351–363.

[47] Farvolden, J. M. and Powell, W. B. (1994), Subgradient methods for the service network design problem, *Transportation Science* **28**(3), 256–272.

[48] Fischetti, M. and Monaci, M. (2009), Light robustness, *in* 'Robust and online large-scale optimization', Springer, pp. 61–84.

[49] Fleischer, L. and Skutella, M. (2007), Quickest flows over time, *SIAM Journal on Computing* **36**(6), 1600–1630.

[50] Ford Jr, L. R. and Fulkerson, D. R. (1958), Constructing maximal dynamic flows from static flows, *Operations research* **6**(3), 419–433.

[51] Ford Jr, L. R. and Fulkerson, D. R. (1962), *Flows in networks*, Princeton university press.

[52] Frangioni, A. and Gendron, B. (2009), 0–1 reformulations of the multicommodity capacitated network design problem, *Discrete Applied Mathematics* **157**(6), 1229–1241.

[53] Gabrel, V., Murat, C. and Thiele, A. (2014), Recent advances in robust optimization: An overview, *European journal of operational research* **235**(3), 471–483.

[54] Gendron, B., Crainic, T., Frangioni, A., Sansó, B. and Soriano, P. (1999), Telecommunications network planning, *Kluwer Academic Publishers* pp. 1–19.

[55] Georghiou, A., Tsoukalas, A. and Wiesemann, W. (2021), On the optimality of affine decision rules in robust and distributionally robust optimization, *Available at Optimization Online* .

[56] Ghamlouche, I., Crainic, T. G. and Gendreau, M. (2003), Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design, *Operations research* **51**(4), 655–667.

[57] Gleeson, J. and Ryan, J. (1990), Identifying minimally infeasible subsystems of inequalities, *ORSA Journal on Computing* **2**(1), 61–63.

[58] Goh, J. and Sim, M. (2010), Distributionally robust optimization and its tractable approximations, *Operations research* **58**(4-part-1), 902–917.

[59] Groß, M. and Skutella, M. (2012), Maximum multicommodity flows over time without intermediate storage, *in* 'European Symposium on Algorithms', Springer, pp. 539–550.

[60] Gurobi Optimization, L. (2021), 'Gurobi optimizer reference manual (v.8.1.1)'. **URL:** *http://www.gurobi.com*

[61] Hall, R. W. (1987), Consolidation strategy: inventory, vehicles and terminals, *Journal of business logistics* **8**(2), 57.

[62] Hewitt, M. (2019), Enhanced dynamic discretization discovery for the continuous time load plan design problem, *Transportation science* **53**(6), 1731–1750.

[63] Hewitt, M., Crainic, T. G., Nowak, M. and Rei, W. (2019), Scheduled service network design with resource acquisition and management under uncertainty, *Transportation Research Part B: Methodological* **128**, 324–343.

[64] Hewitt, M., Nemhauser, G. L. and Savelsbergh, M. W. (2010), Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem, *INFORMS Journal on Computing* **22**(2), 314–325.

[65] Hewitt, M., Nemhauser, G. and Savelsbergh, M. W. (2013), Branch-and-price guided search for integer programs with an application to the multicommodity fixed-charge network flow problem, *INFORMS Journal on Computing* **25**(2), 302–316.

[66] Hoff, A., Lium, A.-G., Løkketangen, A. and Crainic, T. G. (2010), A metaheuristic for stochastic service network design, *Journal of Heuristics* **16**(5), 653–679.

[67] Hosseininasab, A. (2015), The continuous time service network design problem, Master's thesis, University of Waterloo.

[68] Hrušovskỳ, M., Demir, E., Jammernegg, W. and Van Woensel, T. (2018), Hybrid simulation and optimization approach for green intermodal transportation problem with travel time uncertainty, *Flexible Services and Manufacturing Journal* **30**(3), 486–516.

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

[70] Hu, W., Toriello, A. and Dessouky, M. (2018), Integrated inventory routing and freight consolidation for perishable goods, *European Journal of Operational Research* **271**(2), 548–560.

[71] Jalilvand-Nejad, A., Shafaei, R. and Shahriari, H. (2016), Robust optimization under correlated polyhedral uncertainty set, *Computers & Industrial Engineering* **92**, 82–94.

[72] Jarrah, A. I., Johnson, E. and Neubert, L. C. (2009), Large-scale, less-than-truckload service network design, *Operations Research* **57**(3), 609–625.

[73] Koster, A. M., Kutschka, M. and Raack, C. (2013), Robust network design: Formulations, valid inequalities, and computations, *Networks* **61**(2), 128–149.

[74] Kouvelis, P. and Yu, G. (2013), *Robust discrete optimization and its applications*, Vol. 14, Springer Science & Business Media.

[75] Lagos, F., Boland, N. and Savelsbergh, M. (2020), The continuous-time inventory-routing problem, *Transportation Science* **54**(2), 375–399.

[76] Lanza, G., Crainic, T. G., Rei, W. and Ricciardi, N. (2021), Scheduled service network design with quality targets and stochastic travel times, *European Journal of Operational Research* **288**(1), 30–46.

[77] Lee, C.-Y., Shu, S. and Xu, Z. (2021), Optimal global liner service procurement by utilizing liner service schedules, *Production and Operations Management* **30**(3), 703–714.

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

[79] Lium, A.-G., Crainic, T. G. and Wallace, S. W. (2009), A study of demand stochasticity in service network design, *Transportation Science* **43**(2), 144–157.

[80] Long, D. Z., Sim, M. and Zhou, M. (2022), Robust satisficing, *Operations Research* .

[81] Marshall, L., Boland, N., Savelsbergh, M. and Hewitt, M. (2021), Interval-based dynamic discretization discovery for solving the continuous-time service network design problem, *Transportation Science* **55**(1), 29–51.

[82] Medina, J., Hewitt, M., Lehuédé, F. and Péton, O. (2019), Integrating long-haul and local transportation planning: The service network design and routing problem, *EURO Journal on Transportation and Logistics* **8**(2), 119–145.

[83] Mokarami, S. and Hashemi, S. M. (2015), Constrained shortest path with uncertain transit times, *Journal of Global Optimization* **63**(1), 149–163.

[84] Montemanni, R., Barta, J., Mastrolilli, M. and Gambardella, L. M. (2007), The robust traveling salesman problem with interval data, *Transportation Science* **41**(3), 366–381.

[85] Needham, P. M. and Evers, P. T. (1998), The influence of individual cost factors on the use of emergency transshipments, *Transportation Research Part E: Logistics and Transportation Review* **34**(2), 149–160.

[86] Pedersen, M. B., Crainic, T. G. and Madsen, O. B. (2009), Models and tabu search metaheuristics for service network design with asset-balance requirements, *Transportation Science* **43**(2), 158–177.

[87] Remli, N., Amrouss, A., El Hallaoui, I. and 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.

[88] Rudi, A., Fröhling, M., Zimmer, K. and Schultmann, F. (2016), Freight transportation planning considering carbon emissions and in-transit holding costs: a capacitated multi-commodity network flow model, *EURO Journal on Transportation and Logistics* **5**(2), 123–160.

[89] Ruszczyński, A. and Shapiro, A. (2003), Stochastic programming models, *Handbooks in operations research and management science* **10**, 1–64.

[90] Sarayloo, F. (2018), Learning-Based Matheuristic Solution Methods for Stochastic Network Design, PhD thesis, University of Montreal.

[91] Sarayloo, F., Crainic, T. G. and Rei, W. (2021*a*), A learning-based matheuristic for stochastic multicommodity network design, *INFORMS Journal on Computing* **33**(2), 643–656.

[92] Sarayloo, F., Crainic, T. G. and Rei, W. (2021*b*), A reduced cost-based restriction and refinement matheuristic for stochastic network design problem, *Journal of Heuristics* **27**(3), 325–351.

[93] Simchi-Levi, D., Trichakis, N. and Zhang, P. Y. (2019), Designing response supply chain against bioattacks, *Operations Research* **67**(5), 1246–1268.

[94] Skutella, M. (2009), An introduction to network flows over time, *in* 'Research trends in combinatorial optimization', Springer, pp. 451–482.

[95] Sohoni, M., Lee, Y.-C. and Klabjan, D. (2011), Robust airline scheduling under block-time uncertainty, *Transportation Science* **45**(4), 451–464.

[96] Soyster, A. L. (1973), Convex programming with set-inclusive constraints and applications to inexact linear programming, *Operations research* **21**(5), 1154–1157.

[97] Thiele, A., Terry, T. and Epelman, M. (2009), Robust linear optimization with recourse, *Rapport technique* pp. 4–37.

[98] Tyan, J. C., Wang, F.-K. and Du, T. C. (2003), An evaluation of freight consolidation policies in global third party logistics, *Omega* **31**(1), 55–62.

[99] Ulku, M. A. (2009*a*), Analysis of shipment consolidation in the logistics supply chain, PhD thesis, University of Waterloo.

[100] Ülkü, M. A. (2009*b*), Comparison of typical shipment consolidation programs: structural results, *Management Science and Engineering* **3**(4), 27–33.

[101] Van Loon, J. (1981), Irreducibly inconsistent systems of linear inequalities, *European Journal of Operational Research* **8**(3), 283–288.

[102] Van Parys, B. P., Esfahani, P. M. and Kuhn, D. (2021), From data to decisions: Distributionally robust optimization is optimal, *Management Science* **67**(6), 3387–3402.

[103] Vu, D. M., Hewitt, M., Boland, N. and Savelsbergh, M. (2020), Dynamic discretization discovery for solving the time-dependent traveling salesman problem with time windows, *Transportation science* **54**(3), 703–720.

[104] Wang, S. and Meng, Q. (2012*a*), Liner ship route schedule design with sea contingency time and port time uncertainty, *Transportation Research Part B: Methodological* **46**(5), 615–633.

[105] Wang, S. and Meng, Q. (2012*b*), Robust schedule design for liner shipping services, *Transportation Research Part E: Logistics and Transportation Review* **48**(6), 1093–1106.

[106] Wang, X. and Regan, A. C. (2002), Local truckload pickup and delivery with hard time window constraints, *Transportation Research Part B: Methodological* **36**(2), 97–112.

[107] Wang, X. and Regan, A. C. (2009), On the convergence of a new time window discretization method for the traveling salesman problem with time window constraints, *Computers & Industrial Engineering* **56**(1), 161–164.

[108] Wang, Z. and Qi, M. (2020), Robust service network design under demand uncertainty, *Transportation Science* **54**(3), 676–689.

[109] Wieberneit, N. (2008), Service network design for freight transportation: a review, *OR spectrum* **30**(1), 77–112.

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

[111] Zeng, B. and Zhao, L. (2013), Solving two-stage robust optimization problems using a column-and-constraint generation method, *Operations Research Letters* **41**(5), 457–461.

[112] Zhang, Y., Song, S., Shen, Z.-J. M. and Wu, C. (2017), Robust shortest path problem with distributional uncertainty, *IEEE transactions on intelligent transportation systems* **19**(4), 1080–1090.

[113] Zhang, Y., Zhang, Z., Lim, A. and Sim, M. (2021), Robust data-driven vehicle routing with time windows, *Operations Research* **69**(2), 469–485.

[114] Zhao, Y., Xue, Q., Cao, Z. and Zhang, X. (2018), A two-stage chance constrained approach with application to stochastic intermodal service network design problems, *Journal of Advanced Transportation* **2018**.