A New Exact Algorithm for Continuous-Time Service Network Design with Splittable Deliveries

Yarui Zhang
Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, yarui.zhang@connect.polyu.hk

Shengnan Shu
Institute of Operations Research and Analytics, National University of Singapore, shengnan.shu@connect.polyu.hk

Roberto Baldacci
College of Science and Engineering, Hamad Bin Khalifa University, rbaldacci@hbku.edu.qa

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

Abstract. The Continuous-Time Service Network Design Problem (CTSNDP) arises in consolidation-based transportation systems, where shipments share paths in space and time to reduce cost. The CTSNDP determines minimum-cost transportation plans over a continuous-time horizon. To provide more consolidation opportunities and achieve further cost reductions, CTSNDP with Splittable Deliveries (SCTSNDP) allows each commodity to be divided into smaller shipments which may follow different routes and schedules. However, solving SCTSNDP exactly is computationally challenging. A direct formulation based on a complete time discretization is typically too large to be solved efficiently. Alternatively, CTSNDP with Unsplittable Deliveries (USCTSNDP) provides a restriction for SCTSNDP when each multi-unit commodity is replaced by predefined multiple sub-commodities. However, such construction substantially enlarges the commodity set and limits the effectiveness of the existing dynamic discretization discovery (DDD) algorithms. To address this challenge, we establish feasibility and optimality equivalences between SCTSNDP and the solution-induced USCTSNDP counterparts, and develop a new DDD algorithm for the SCTSNDP, which iteratively derives upper bounds and refines time discretization by leveraging established DDD components for USCTSNDP. Computational results show that the proposed method outperforms benchmark approaches for SCTSNDP. Extensive experiments further show that splittable deliveries yield significant cost savings with hight time flexibility and tight relative network capacity.

Key words: Service network design, Splittable delivery, Continuous time, Dynamic discretization discovery, Exact algorithm

1. Introduction

The Service Network Design Problem (SNDP) aims to select transportation activities and allocate resources over a planning horizon to deliver goods profitably (Crainic 2000, Crainic and Hewitt 2021). It typically arises in consolidation-based transportation systems operated by less-than-truckload (LTL) and small-parcel delivery carriers. In these systems, multiple commodities with different origins and destinations may be consolidated into the same vehicle dispatch over shared segments of their routes to reduce vehicle-based transportation costs (Hewitt et al. 2019). Such consolidation, however, is possible only when commodities are compatible not only spatially

but also temporally, making scheduling decisions critical to SNDP. These scheduling decisions are often modeled through time discretization (Crainic et al. 2016), and the chosen time granularity can strongly affect which consolidation opportunities are represented. To avoid losing or distorting such opportunities due to coarse discretization, the Continuous-Time SNDP (CTSNDP) has been proposed to model scheduling decisions over a continuous-time planning horizon and thereby capture potential consolidations more accurately (Boland et al. 2017).

In the CTSNDP, commodities are routed and consolidated through a physical network over a continuous-time planning horizon with the objective of minimizing total transportation cost. This cost consists of fixed costs incurred by operating vehicle dispatches on transportation service segments and flow costs determined by shipment quantities and unit flow costs. By optimizing decisions in continuous time, the CTSNDP identifies a transportation plan that specifies commodity routing, shipment consolidations, and service dispatch schedules. Recent literature on the CTSNDP has commonly focused on settings in which the demand for each commodity must be transported from its origin to its destination along a single route, without splittable deliveries (see, e.g., (Boland et al. 2017, Marshall et al. 2021, Shu et al. 2024)). We refer to this standard setting as the Continuous-Time SNDP with Unsplittable Deliveries (USCTSNDP).

To obtain the continuous-time optimal solutions for the USCTSNDP, Boland et al. (2017) proposed a dynamic discretization discovery (DDD) algorithm. Rather than solving the problem directly on the complete time-expanded network containing all relevant time points, the DDD algorithm starts with a coarse time discretization, solves the resulting relaxation on the corresponding partial network, and iteratively refines the discretization by adding new time points only when they are needed to prove optimality. This approach provides a significant computational advantage over directly solving the problem on the complete time-expanded network. Following Boland et al. (2017), several studies have improved the DDD algorithm through enhanced initial relaxation, upper bound computation, and discretization refinement strategies (Marshall et al. 2021, Van Dyk and Koenemann 2024, Shu et al. 2025). The DDD algorithm has also been extended to some variants of the USCTSNDP with predefined paths (Hewitt 2019), flexible available and due times (Hewitt 2022), and holding costs (Shu et al. 2024), as well as to other transportation problems (van Lieshout and van der Schaft 2026, He et al. 2022, Ocampo-Giraldo et al. 2025).

However, the unsplittable delivery assumption may limit consolidation opportunities, as assigning the entire demand for a commodity to a single route and schedule prevents different portions of that demand from being consolidated across different dispatches. Splittable deliveries relax this

restriction by allowing a commodity to be divided into smaller shipments that may follow different routes and schedules, providing more opportunities for consolidations and reducing transportation costs (Sherali et al. 2000, Atamtürk and Rajan 2002, Archetti et al. 2008, 2016, Gouveia et al. 2023). For example, in vehicle routing problems, allowing splittable deliveries has been shown to yield cost savings of up to 50% and to reduce the number of delivery routes needed to satisfy all demand (Dror and Trudeau 1989, Archetti et al. 2006). Motivated by this context, we investigate the Continuous-Time SNDP with Splittable Deliveries (or SCTSNDP in short) in this study.

Despite the benefit of splittable deliveries, the SCTSNDP has received limited attention and remains computationally challenging. One straightforward approach to the SCTSNDP is to solve the problem over a complete time-expanded network. Similar to the USCTSNDP, the complete time discretization can impose a substantial computational burden. Alternatively, following the freight-splitting interpretation in Marshall et al. (2021), splittable deliveries can be tackled by breaking commodities into smaller pieces. However, finer demand splitting enlarges the number of commodities, leading to a significant increase in model size and undermining computational efficiency of the existing DDD algorithm. These limitations motivate the development of an exact method to efficiently and effectively solve the SCTSNDP.

1.1. Contributions of this paper

To address the limitations of existing methods for the SCTSNDP, we extend the DDD framework from unsplittable to splittable deliveries, avoiding both the complete time-expanded network and the explicit enumeration of commodity demand splits. Our main contributions are as follows:

- We propose valid relaxations for the SCTSNDP on the same partially time-expanded networks as those used for the corresponding USCTSNDP instances, which are defined on the same underlying network and commodity set.
- For any SCTSNDP relaxation solution, we construct the corresponding solution-induced USCTSNDP instance, relaxation, and mapped solution, and establish feasibility and optimality equivalences between the original and mapped solutions.
- We develop a new DDD algorithm for the SCTSNDP that avoids pre-enumerating commodity demand splitting and complete time discretization, instead iteratively mapping SCTSNDP relaxation solutions to solution-induced USCTSNDP counterparts to support optimality checking and refine the time discretization.
- We validate the efficiency and effectiveness of the new DDD algorithm, as well as the benefits of allowing splittable deliveries in the CTSNDP, through extensive computational experiments.

The results show that the proposed method significantly outperforms existing approaches for the SCTSNDP, while delivery splitting yields greater cost savings under tight network capacity and flexible commodity time windows.

The remainder of the paper is organized as follows. Section 2 defines the SCTSNDP, presents two time-indexed formulations and solution approaches, and discusses their limitations. Section 3 introduces an SCTSNDP relaxation and establishes feasibility and optimality conditions using solution-induced USCTSNDP counterparts. Section 4 presents the proposed algorithm, Section 5 reports the computational results, and Section 6 concludes the paper.

2. Problem Description, Formulations, and Tractability Bottleneck

This section defines the SCTSNDP in Section 2.1, presents two time-indexed formulations in Sections 2.2 and 2.3, and reviews the associated solution methods and their limitations in Section 2.4, motivating our proposed method.

2.1. Problem Description

The SCTSNDP is a continuous-time variant of the SNDP that allows splittable deliveries and can be described by extending the setting of the USCTSNDP presented in Boland et al. (2017).

The SCTSNDP takes as input a directed graph $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ represents a set of nodes and $\mathcal{A}$ denotes a set of arcs, and a set of commodities $\mathcal{K}$ that must be transported through the network $\mathcal{D}$. The network $\mathcal{D}$ is regarded as a *flat network*, where the nodes in $\mathcal{N}$ correspond to physical terminals. Each arc $a = (i, j) \in \mathcal{A}$ is associated with a travel time $\tau_{ij} \in \mathbb{N}_{>0}$, a per-unit flow cost $c_{ij}^k \in \mathbb{R}_{\geq 0}$ for each commodity $k \in \mathcal{K}$, while each vehicle dispatch on the arc involves a fixed cost $f_{ij} \in \mathbb{R}_{>0}$ and a capacity $u_{ij} \in \mathbb{N}_{>0}$. Each commodity $k \in \mathcal{K}$ has a demand $q^k \in \mathbb{R}_{>0}$ that must be delivered from its origin $o^k \in \mathcal{N}$ to its destination $d^k \in \mathcal{N}$ within a time window $[e^k, l^k]$, where $e^k \in \mathbb{N}_{\geq 0}$ is the time when commodity k becomes available at its origin, and $l^k \in \mathbb{N}_{>0}$ is the time it is due at its destination. Similar to the USCTSNDP, in the SCTSNDP, multiple flows of different commodities can be consolidated into a single vehicle dispatch on arc $(i, j) \in \mathcal{A}$, thereby sharing the fixed cost for using the vehicle. However, unlike the USCTSNDP, the SCTSNDP allows a commodity to follow multiple paths and schedules from its origin to its destination when beneficial. The objective of the SCTSNDP is to determine the optimal flow splits for all commodities, including their delivery quantity, delivery paths, and dispatch schedules, as well as the consolidation plan of these flow splits, so as to minimize the sum of total fixed and variable flow costs.

To represent a feasible solution of the SCTSNDP, we define M_k as the set of *flow splits* for each commodity $k \in \mathcal{K}$. For each flow split $m \in M_k$, we define a triple $w^{k,m} = (g^{k,m}, p^{k,m}, t^{k,m})$

to represent its flow amount, delivery path, and dispatch times. Specifically, the first element $g^{k,m}$ represents the amount of commodity k assigned to flow split m. The second element $p^{k,m}$ denotes the delivery path of flow split m in the flat network. It consists of a sequence of arcs $a_h^{k,m} \in \mathcal{A}$, i.e.,$p^{k,m} = (a_1^{k,m}, \ldots, a_{|p^{k,m}|}^{k,m})$, and is referred to as a *flat path*. The delivery path $p^{k,m}$ is called *k-feasible* for commodity $k \in \mathcal{K}$, if it is a flat path from o^k to d^k with $e^k + \sum_{a \in p^{k,m}} \tau_a \leq l^k$. We assume that there always exists a *k-feasible flat path* for each $k \in \mathcal{K}$ which is sufficient to ensure a feasible solution to the problem. Moreover, $t^{k,m} = (t^{k,m}_{a_1^{k,m}}, \ldots, t^{k,m}_{a_{|p^{k,m}|}^{k,m}})$ specifies the dispatch times along the delivery path $p^{k,m}$, where $t^{k,m}_{a_1^{k,m}} \geq e^k, t^{k,m}_{a_n^{k,m}} \geq t^{k,m}_{a_{n-1}^{k,m}} + \tau_{a_{n-1}^{k,m}}$ for each $n = 2, \ldots, |p^{k,m}|$, and $t^{k,m}_{a_{|p^{k,m}|}^{k,m}} + \tau_{a_{|p^{k,m}|}^{k,m}} \leq l^k$. Therefore, a feasible solution of the SCTSNDP can be represented as $\mathcal{W} = \{\{w^{k,m}\}_{m \in M_k}\}_{k \in \mathcal{K}}$ with $\sum_{m \in M_k} g^{k,m} = q^k$ for all $k \in \mathcal{K}$.

As in the USCTSNDP, without loss of optimality, all flow splits that pass through the same arc with the same dispatch time can be consolidated. Associated with the solution $\mathcal{W}$, for each arc $a = (i, j) \in \cup_{k \in \mathcal{K}, m \in M_k} p^{k,m}$, we define $\mathcal{J}_a = \{(k, m) : a \in p^{k,m}\}$ as the set of commodity-split pairs traversing arc a. Let $\mu(a) = \{t_a^{k,m} : (k, m) \in \mathcal{J}_a\}$ be the set of distinct dispatch times associated with these flow splits on arc a. Accordingly, for each dispatch time $t \in \mu(a)$, the consolidation set is defined as $\kappa(a, t) = \{(k, m) \in \mathcal{J}_a : t_a^{k,m} = t\}$. Collectively, these sets define how the flow splits are consolidated on arc a.

Therefore, the cost $z(\mathcal{W})$ of a SCTSNDP solution $\mathcal{W}$ can be calculated as :

$$z(\mathcal{W}) = \sum_{a \in \cup_{k \in \mathcal{K}, m \in M_k} p^{k,m}} \sum_{t \in \mu(a)} f_a \Big\lceil \sum_{(k,m) \in \kappa(a,t)} g^{k,m}/u_a \Big\rceil + \sum_{k \in \mathcal{K}} \sum_{m \in M_k} \sum_{a \in p^{k,m}} c_a^k g^{k,m},$$

where the first and second terms represent the fixed and variable costs, respectively. The SCTSNDP seeks a feasible solution $\mathcal{W}$ with minimum total cost $z(\mathcal{W})$. Note that we assume that holding commodities at terminals incurs no additional costs. Moreover, we note that the labels m assigned to the flow splits in M_k are arbitrary; any consistent relabeling leaves the solution unchanged.

2.2. A Continuous-flow Time-Indexed Formulation for the SCTSNDP

Similar to the USCTSNDP, the SCTSNDP can be modeled using a time-indexed formulation based on a time-expanded network (see, e.g., Boland et al. 2017, Marshall et al. 2021, Shu et al. 2025).

We first construct a time-expanded network $\mathcal{D}_\mathcal{T}^\Delta = (\mathcal{N}_\mathcal{T}^\Delta, \mathcal{A}_\mathcal{T}^\Delta \cup \mathcal{H}_\mathcal{T}^\Delta)$ based on a discretization unit Δ, where for each terminal $i \in \mathcal{N}$, the set of discrete time points $\mathcal{T}_i = \{t_1^i, \ldots, t_{|\mathcal{T}_i|}^i\}$ is defined by $\{0, \Delta, \ldots, \Delta \cdot \max_{k \in \mathcal{K}} \lceil l^k/\Delta \rceil\}$ and $\mathcal{T} = \{\mathcal{T}_i\}_{i \in \mathcal{N}}$. The node set $\mathcal{N}_\mathcal{T}^\Delta$ comprises timed-nodes (i, t) for each terminal $i \in \mathcal{N}$ and each time point $t \in \mathcal{T}_i$. The arc set of $\mathcal{D}_\mathcal{T}^\Delta$ consists of the set of holding

arcs $\mathcal{H}_{\mathcal{T}}^{\Delta}$ and the set of service arcs $\mathcal{A}_{\mathcal{T}}^{\Delta}$. Holding arcs $((i,t_n^i),(i,t_{n+1}^i)) \in \mathcal{H}_{\mathcal{T}}^{\Delta}$ are defined for every terminal $i \in \mathcal{N}$ and time index $n \in \{1,\ldots,|\mathcal{T}_i|-1\}$ to represent a dwell time of $(t_{n+1}^i - t_n^i)$ units. Meanwhile, service arcs $((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}$ are constructed for each physical arc $(i,j) \in \mathcal{A}$ and each pair of timed nodes $(i,t),(j,\bar{t}) \in \mathcal{N}_{\mathcal{T}}^{\Delta}$ with $\bar{t} = t + \Delta\lceil \tau_{ij}/\Delta \rceil$ to model a dispatch from terminal i at time t that reaches terminal j at time $\bar{t}$.

Based on the network $\mathcal{D}_{\mathcal{T}}^{\Delta}$, we present a time-indexed formulation for the SCTSNDP under the commodity set $\mathcal{K}$, referred to as SCTSNDP($\mathcal{K}$). Let $x_{ij}^{kt\bar{t}}$ be a continuous variable from zero to one to represent the fraction of demand for commodity $k \in \mathcal{K}$ that travels from i to j departing at time t and arriving at $\bar{t}$. Let $y_{ij}^{t\bar{t}}$ be a nonnegative integer variable representing the number of vehicles on arc $(i,j) \in \mathcal{A}$ required to transport the consolidated demand dispatched from terminal i at time t and arriving at terminal j at time $\bar{t}$. Accordingly, SCTSNDP($\mathcal{K}$) can be formulated as the following continuous-flow time-indexed mixed integer programming model SSND($\mathcal{D}_{\mathcal{T}}^{\Delta},\mathcal{K}$):

$$z(\mathcal{D}_{\mathcal{T}}^{\Delta},\mathcal{K}) = \min \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} f_{ij} y_{ij}^{t\bar{t}} + \sum_{k\in\mathcal{K}} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} (c_{ij}^k q^k) x_{ij}^{kt\bar{t}} \tag{1}$$

$$s.t. \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}\cup\mathcal{H}_{\mathcal{T}}^{\Delta}} x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}^{\Delta}\cup\mathcal{H}_{\mathcal{T}}^{\Delta}} x_{ji}^{k\bar{t}t}$$

$$= \begin{cases} 1 & (i,t) = (o^k, \min\{s \in \mathcal{T}_{o^k} : s \geq e^k\}), \\ -1 & (i,t) = (d^k, \max\{s \in \mathcal{T}_{d^k} : s \leq l^k\}), \\ 0 & \text{otherwise}, \end{cases} \quad \forall k \in \mathcal{K}, (i,t) \in \mathcal{N}_{\mathcal{T}}^{\Delta} \tag{2}$$

$$\sum_{k\in\mathcal{K}} q^k x_{ij}^{kt\bar{t}} \leq u_{ij} y_{ij}^{t\bar{t}}, \quad \forall ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}, \tag{3}$$

$$x_{ij}^{kt\bar{t}} \in [0,1], \quad \forall k \in \mathcal{K}, ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta} \cup \mathcal{H}_{\mathcal{T}}^{\Delta}, \tag{4}$$

$$y_{ij}^{t\bar{t}} \in \mathbb{N}_{\geq 0}, \quad \forall ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta}. \tag{5}$$

The objective function (1) aims to minimize the total cost, with the first and second terms representing fixed and variable costs, respectively. For any feasible solution $(\boldsymbol{x},\boldsymbol{y})$, let $z(\boldsymbol{x},\boldsymbol{y})$ denote its objective value. Constraints (2) enforce flow conservation to ensure that the demand for each commodity is met, departing from its origin upon availability and reaching its destination by the specified due time. Constraints (3) ensure that sufficient transportation capacity is provided by the vehicles to accommodate the total consolidated demand flowing from terminal i at time t to terminal j at time $\bar{t}$. Constraints (4) and (5) specify the domains of the decision variables. We note that an optimal solution $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ may be assumed *timed-cycle-free*, meaning that every commodity's

positive-flow timed arcs contain no directed cycle in the time-expanded network $\mathcal{D}^{\Delta}_{\mathcal{T}}$, since any zero-cost cycle can be removed without affecting feasibility or optimality.

Under these constraints, the flow of each commodity $k \in \mathcal{K}$ in any timed-cycle-free feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ to SSND$(\mathcal{D}^{\Delta}_{\mathcal{T}}, \mathcal{K})$ can be decomposed into origin-to-destination timed-path flows. For any fixed such decomposition, let $M_k(\boldsymbol{x}, \boldsymbol{y})$ be the corresponding flow splits, where each split $m \in M_k(\boldsymbol{x}, \boldsymbol{y})$ is characterized by a flow volume $g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ and a delivery timed path $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ from $(o^k, \min\{s \in \mathcal{T}_{o^k} : s \geq e^k\})$ to $(d^k, \max\{s \in \mathcal{T}_{d^k} : s \leq l^k\})$ over $\mathcal{D}^{\Delta}_{\mathcal{T}}$. Each such timed path comprises a flat path $p^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ and an associated schedule of departure times $t^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ for its constituent service arcs. Therefore, each such decomposition yields a feasible solution $\mathcal{W} = \{\{(g^{k,m}(\boldsymbol{x}, \boldsymbol{y}), p^{k,m}(\boldsymbol{x}, \boldsymbol{y}), t^{k,m}(\boldsymbol{x}, \boldsymbol{y}))\}_{m \in M_k(\boldsymbol{x}, \boldsymbol{y})}\}_{k \in \mathcal{K}}$ for SCTSNDP$(\mathcal{K})$, with the same delivery quantities, delivery paths, and dispatch schedules for all commodities, thereby inducing the same consolidations on each arc. The resulting solution satisfies $z(\mathcal{W}) \leq z(\boldsymbol{x}, \boldsymbol{y})$, with equality whenever $(\boldsymbol{x}, \boldsymbol{y})$ is optimal. Moreover, for each timed arc $a \in \mathcal{A}^{\Delta}_{\mathcal{T}}$, we define $\mathcal{J}_a(\boldsymbol{x}, \boldsymbol{y}, M_k(\boldsymbol{x}, \boldsymbol{y})) = \left\{(k, m) : k \in \mathcal{K},\ m \in M_k(\boldsymbol{x}, \boldsymbol{y}),\ a \in P^{k,m}(\boldsymbol{x}, \boldsymbol{y})\right\}$ as the set of flow splits in $M_k(\boldsymbol{x}, \boldsymbol{y})$ traversing timed arc a. The corresponding routing and consolidation plans can then be represented, respectively, by $\mathcal{P}(\boldsymbol{x}, \boldsymbol{y}, M_k(\boldsymbol{x}, \boldsymbol{y})) = \{p^{k,m}(\boldsymbol{x}, \boldsymbol{y})\}_{k \in \mathcal{K}, m \in M_k(\boldsymbol{x}, \boldsymbol{y})}$ and $C(\boldsymbol{x}, \boldsymbol{y}, M_k(\boldsymbol{x}, \boldsymbol{y})) = \{((i, j), \mathcal{J}_a(\boldsymbol{x}, \boldsymbol{y}, M_k(\boldsymbol{x}, \boldsymbol{y}))) : a = ((i, t), (j, \bar{t})) \in \mathcal{A}^{\Delta}_{\mathcal{T}}\}$. Together, $(\mathcal{P}(\boldsymbol{x}, \boldsymbol{y}, M_k(\boldsymbol{x}, \boldsymbol{y})), C(\boldsymbol{x}, \boldsymbol{y}, M_k(\boldsymbol{x}, \boldsymbol{y})))$ defines a *flat solution* to SCTSNDP$(\mathcal{K})$.

The quality of an optimal solution to SSND$(\mathcal{D}^{\Delta}_{\mathcal{T}}, \mathcal{K})$ is influenced by the discretization parameter Δ, as it introduces deviations to travel times, available times, and due times. Therefore SSND$(\mathcal{D}^{\Delta}_{\mathcal{T}}, \mathcal{K})$ is a restriction of SCTSNDP$(\mathcal{K})$, and its optimal objective value provides an upper bound for the original SCTSNDP$(\mathcal{K})$. However, as noted by Boland et al. (2017), these deviations disappear when τ_{ij}/Δ, e^k/Δ, and l^k/Δ are all integers (i.e., when Δ is a common divisor of all temporal parameters). Let $\hat{\Delta}$ denote the greatest common divisor of all τ_{ij}, e^k and l^k, where $\hat{\Delta} = 1$ in the worst case. We denote the corresponding discrete time set $\hat{\mathcal{T}}$ as the *complete time discretization* and the resulting time-expanded network $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ is referred to as a *complete time-expanded network*. Consequently, SSND$(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K})$ is a *complete continuous-flow time-indexed model* for SCTSNDP$(\mathcal{K})$, where any optimal solution $(\boldsymbol{x}, \boldsymbol{y})$ to the SSND$(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K})$ corresponds to an optimal solution $\mathcal{W} = \{\{(g^{k,m}(\boldsymbol{x}, \boldsymbol{y}), p^{k,m}(\boldsymbol{x}, \boldsymbol{y}), t^{k,m}(\boldsymbol{x}, \boldsymbol{y}))\}_{m \in M_k(\boldsymbol{x}, \boldsymbol{y})}\}_{k \in \mathcal{K}}$ for SCTSNDP$(\mathcal{K})$ with the same total cost.

2.3. A Binary-flow Time-indexed Formulation for the SCTSNDP

As noted by Marshall et al. (2021), SCTSNDP$(\mathcal{K})$ can be tackled heuristically by splitting commodity demands into smaller pieces a priori and solving the resulting USCTSNDP instance. For clarity, we first introduce the following definitions.

DEFINITION 1. Given a commodity set $\mathcal{K}$, for each $k \in \mathcal{K}$, let $\Gamma_k = \{\gamma_1^k, \gamma_2^k, \ldots, \gamma_{|\Gamma_k|}^k\}$ denote a finite set of sub-commodities derived from k, where each sub-commodity $\gamma_m^k \in \Gamma_k$ is a *copy* of k that inherits its origin, destination, earliest available time, due time, and unit flow cost from k, i.e., $o^{\gamma_m^k} = o^k$,$d^{\gamma_m^k} = d^k$, $e^{\gamma_m^k} = e^k$, $l^{\gamma_m^k} = l^k$, and $c_a^{\gamma_m^k} = c_a^k, \forall a \in \mathcal{A}$, and the total demand satisfies $\sum_{m=1}^{|\Gamma_k|} q^{\gamma_m^k} = q^k, q^{\gamma_m^k} > 0$. We define such Γ_k as a *feasible demand disaggregation* for commodity k.

DEFINITION 2. Given a feasible demand disaggregation $\Gamma_k = \{\gamma_m^k : m = 1, \ldots, |\Gamma_k|\}$ for each $k \in \mathcal{K}$, we define $\Phi_{\mathcal{K}} = \bigcup_{k\in\mathcal{K}} \Gamma_k$ as a *feasible split* of the commodity set $\mathcal{K}$.

Analogously, we use USCTSNDP($\mathcal{K}$) to denote the USCTSNDP associated with commodity set $\mathcal{K}$. Given a feasible split $\Phi_{\mathcal{K}}$ of $\mathcal{K}$, the corresponding USCTSNDP($\Phi_{\mathcal{K}}$) is defined on the same underlying network $\mathcal{D}$ and constitutes a restriction of the original SCTSNDP($\mathcal{K}$) under which demand can be divided only according to the predefined split. As noted by Boland et al. (2017), the resulting USCTSNDP($\Phi_{\mathcal{K}}$) can also be formulated on the time-expanded network $\mathcal{D}_{\mathcal{T}}^{\Delta}$, yielding the model USSND($\mathcal{D}_{\mathcal{T}}^{\Delta}, \Phi_{\mathcal{K}}$). This model shares the same objective function and constraints as SSND($\mathcal{D}_{\mathcal{T}}^{\Delta}, \Phi_{\mathcal{K}}$), except that the flow variables $x_{ij}^{kt\bar{t}}$ are binary: $x_{ij}^{kt\bar{t}} \in \{0,1\}, \forall k \in \Phi_{\mathcal{K}}, ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}^{\Delta} \cup \mathcal{H}_{\mathcal{T}}^{\Delta}$. Here, the binary variable $x_{ij}^{kt\bar{t}}$ indicates whether sub-commodity $k \in \Phi_{\mathcal{K}}$ is routed along arc (i,j) with departure time t and arrival time $\bar{t}$, thereby ensuring each sub-commodity is routed along a single delivery path and assigned a unique schedule. An optimal solution to USCTSNDP($\Phi_{\mathcal{K}}$) can be obtained by solving the complete binary-flow time-indexed model USSND($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}, \Phi_{\mathcal{K}}$), and, by Proposition 1, its objective value provides an upper bound for the original SCTSNDP($\mathcal{K}$). Refining the predefined split with smaller demand-splitting granularity can tighten this upper bound by allowing the demand of each original commodity to be divided among routes and schedules in more flexible proportions. Indeed, as established in Corollary 1, there exists a feasible split $\bar{\Phi}_{\mathcal{K}}$ such that solving USCTSNDP($\bar{\Phi}_{\mathcal{K}}$) yields an optimal solution to SCTSNDP($\mathcal{K}$), although such a split is generally unknown a priori.

PROPOSITION 1. *Given any feasible split $\Phi_{\mathcal{K}}$, every feasible solution to USCTSNDP($\Phi_{\mathcal{K}}$) corresponds to a feasible solution to SCTSNDP($\mathcal{K}$) with an identical cost.*

COROLLARY 1. *There exists a finite feasible split $\bar{\Phi}_{\mathcal{K}}$ such that USCTSNDP($\bar{\Phi}_{\mathcal{K}}$) and SCTSNDP($\mathcal{K}$) have the same optimal cost.*

2.4. Tractability Bottleneck

For SCTSNDP($\mathcal{K}$) studied, Section 2.2 presents an exact continuous-flow time-indexed formulation SSND($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}, \mathcal{K}$), whereas Section 2.3 presents a binary-flow time-indexed formulation

USSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \Phi_{\mathcal{K}}$), for the restriction induced by a given feasible split $\Phi_{\mathcal{K}}$. In this section, we review existing solution methodologies based on these formulations and analyze their limitations.

The continuous-flow time-indexed formulation, SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$), is defined on the complete time-expanded network $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$, where solving the model directly yields the optimal solution for SCTSNDP($\mathcal{K}$). In practice, $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ tends to be prohibitively large, and the corresponding complete time-indexed model SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$) is computationally challenging. To reduce the size of the complete time-indexed model, Boland et al. (2017) showed that only a subset of time points is required in the discretization to preserve optimality. This reduction can be applied to the SCTSNDP, as stated in Proposition 2, under the assumption that holding freight at terminals incurs no additional cost. Nevertheless, this reduced model remains too large to render the SCTSNDP computationally tractable, and no alternative solution methods or specialized algorithms currently exist beyond direct optimization to solve SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$) efficiently.

PROPOSITION 2. *For any general time-expanded network $\mathcal{D}_{\mathcal{T}}$ with $\mathcal{T} \in \hat{\mathcal{T}}$, in which service arcs are of the form $((i,t),(j,t+\tau_{ij}))$ and holding arcs connect consecutive time points in each $\mathcal{T}_i$, to ensure that any optimal solution to SSND($\mathcal{D}_{\mathcal{T}}, \mathcal{K}$) is an optimal solution to SCTSNDP($\mathcal{K}$), it is sufficient for $\mathcal{T}$ to consist of time points of e^k for some commodity $k \in \mathcal{K}$ or $e^k + \sum_{a \in p} \tau_a$ for some commodity k and some walk $p \subseteq \mathcal{A}$ originating at o^k.*

Alternatively, one could solve the binary-flow time-indexed formulation USSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \Phi_{\mathcal{K}}$) with a feasible split $\Phi_{\mathcal{K}}$ to obtain a heuristic solution to SCTSNDP($\mathcal{K}$) by using standard DDD algorithms for the USCTSNDP (Boland et al. 2017, Marshall et al. 2021, Shu et al. 2025). Although these algorithms avoid explicitly handling the complete time-expanded network, the resulting heuristic solution may provide a loose upper bound because the optimality-preserving finite split whose existence is established by Corollary 1 is generally unknown a priori. Using finer splits can tighten this upper bound but rapidly enlarges the binary-flow formulation, making it increasingly difficult to solve even with these DDD algorithms.

Together, these limitations motivate the new exact method developed in the following sections, which avoids both complete time discretization and prespecified demand splitting.

3. Relaxation and USCTSNDP Counterparts of the SCTSNDP

In this section, we introduce a relaxation of the SCTSNDP and, for any solution to this relaxation, construct a mapped USCTSNDP instance, its relaxation, and the corresponding mapped solution. We establish feasibility and optimality equivalences between the given relaxation solution and

its mapped counterpart, providing the theoretical foundation for the exact method in Section 4. Appendix EC.1 summarizes the relationships among the problem variants, formulations, relaxations, and mapped counterparts.

3.1. Partially Time-expanded Network and Relaxation

As discussed in Section 2.2, the SCTSNDP can be modeled using the continuous-flow time-indexed formulation on the complete time-expanded network SSND($\mathcal{D}^{\hat{\mathbb{A}}}_{\hat{\mathcal{T}}}, \mathcal{K}$). However, even after the reduction of time points established in Proposition 2, the resulting time-expanded network may remain prohibitively large. A similar computational challenge arises in the USCTSNDP, for which Boland et al. (2017) introduced the concept of a *partially time-expanded network*. Based on this network, they derive a relaxation of the original USCTSNDP and propose the DDD algorithm for the USCTSNDP, which dynamically refines the partially time-expanded network to iteratively strengthen the relaxation. Next, we review this partially time-expanded network concept in the context of the USCTSNDP, and subsequently adapt it to develop a relaxation for the SCTSNDP.

A partially time-expanded network, denoted by $\underline{\mathcal{D}_{\mathcal{T}}}$, is constructed based on a small subset of the time points of the complete time discretization $\hat{\mathcal{T}}$ and contains significantly fewer arcs with shorter travel times than the complete time-expanded network. The standard partially time-expanded network, represented as $\underline{\mathcal{D}_{\mathcal{T}}} = (\underline{\mathcal{N}_{\mathcal{T}}}, \underline{\mathcal{A}_{\mathcal{T}}} \cup \underline{\mathcal{H}_{\mathcal{T}}})$, is first introduced by Boland et al. (2017). Specifically, $\mathcal{T} \subset \hat{\mathcal{T}}$ such that $|\mathcal{T}_i| \ll |\hat{\mathcal{T}}_i|$ for each $i \in \mathcal{N}$, where each $t \in \mathcal{T}_i$ corresponds to a node $(i,t) \in \underline{\mathcal{N}_{\mathcal{T}}}$. The holding arcs in $\underline{\mathcal{H}_{\mathcal{T}}}$ link the timed-nodes of two consecutive time points at the same terminal and the service arcs in $\underline{\mathcal{A}_{\mathcal{T}}}$ are constructed to satisfy the following two properties: (i) every arc $((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}$ satisfies $\bar{t} \le t + \tau_{ij}$; (ii) for every $(i,j) \in \mathcal{A}$ and every $(i,t) \in \underline{\mathcal{N}_{\mathcal{T}}}$, there exists a timed-copy arc $((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}$ such that no node $(j,t') \in \underline{\mathcal{N}_{\mathcal{T}}}$ exists where $\bar{t} < t' \le t + \tau_{ij}$. Boland et al. (2017) proved that the time-indexed formulation USSND($\underline{\mathcal{D}_{\mathcal{T}}}$), based on any such partially time-expanded network $\underline{\mathcal{D}_{\mathcal{T}}}$, provides a relaxation of the USCTSNDP.

Other types of partially time-expanded networks have also been proposed for the USCTSNDP to tighten the relaxation while minimizing network size. For example, Marshall et al. (2021) introduced interval-based partially time-expanded commodity networks, whereas Shu et al. (2025) developed time-node-based commodity networks, both by eliminating commodity-specific arcs that are temporally infeasible. In this paper, we let $\underline{\mathcal{D}_{\mathcal{T}}} = (\underline{\mathcal{N}_{\mathcal{T}}}, \underline{\mathcal{A}_{\mathcal{T}}} \cup \underline{\mathcal{H}_{\mathcal{T}}})$ generically denote any such partially time-expanded network to simplify the subsequent discussion.

Although these partially time-expanded networks were originally developed for the USCTSNDP, the following theorem shows that any such network that yields a valid relaxation for the USCTSNDP also provides a valid relaxation for the SCTSNDP.

THEOREM 1. *Given any partially time-expanded network* $\underline{\mathcal{D}_\mathcal{T}}$ *such that* USSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$) *is a relaxation of* USCTSNDP($\mathcal{K}$)*, in the sense that every optimal solution to* USCTSNDP($\mathcal{K}$) *corresponds to a feasible solution of* USSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$) *with no greater cost, then* SSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$) *is a relaxation of* SCTSNDP($\mathcal{K}$).

3.2. Mapping to Solution-Induced USCTSNDP Counterparts

Given any timed-cycle-free feasible solution to the relaxation SSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$) of SCTSNDP($\mathcal{K}$), we map the problem, relaxation, and solution to their corresponding USCTSNDP counterparts.

We begin by constructing a feasible split for the original commodity set $\mathcal{K}$ based on the given timed-cycle-free relaxation solution $(\boldsymbol{x}, \boldsymbol{y})$. For each commodity $k \in \mathcal{K}$, based on the definitions in Section 2.2, $\Gamma_k(\boldsymbol{x}, \boldsymbol{y})$, defined as $\{\gamma_m^k\}_{m \in M_k(\boldsymbol{x}, \boldsymbol{y})}$ where each γ_m^k is a copy of k and with $q^{\gamma_m^k} = g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, is a feasible demand disaggregation for commodity k, and $\Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y}) = \cup_{k \in \mathcal{K}} \Gamma_k(\boldsymbol{x}, \boldsymbol{y})$ yields a feasible split of $\mathcal{K}$. Based on this, we obtain the mapped USCTSNDP instance, defined as USCTSNDP($\Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$) over the same flat network $\mathcal{D}$. As indicated by Proposition 1, any feasible solution to the USCTSNDP($\Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$) is associated with a feasible solution to SCTSNDP($\mathcal{K}$) with the same cost. Given the current partially time-expanded network $\underline{\mathcal{D}_\mathcal{T}}$, we then formulate a relaxation of USCTSNDP($\Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$) using the binary-flow formulation introduced in Section 2.3. This relaxation is denoted by USSND($\underline{\mathcal{D}_\mathcal{T}}, \Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$), and a corresponding feasible solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is constructed as follows:

$$\bar{x}_{ij}^{\gamma_m^k t\bar{t}} = \begin{cases} 1, & \text{if } ((i,t),(j,\bar{t})) \in P^{k,m}(\boldsymbol{x}, \boldsymbol{y}), \\ 0, & \text{otherwise,} \end{cases} \quad \forall k \in \mathcal{K}, m \in M_k(\boldsymbol{x}, \boldsymbol{y}), ((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_\mathcal{T}} \cup \underline{\mathcal{H}_\mathcal{T}} \tag{6}$$

$$\bar{y}_{ij}^{t\bar{t}} = y_{ij}^{t\bar{t}}, \quad \forall ((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_\mathcal{T}}. \tag{7}$$

As established in Lemma 1, the derived solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ has the same cost as $(\boldsymbol{x}, \boldsymbol{y})$ and is optimal for USSND($\underline{\mathcal{D}_\mathcal{T}}, \Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$) whenever $(\boldsymbol{x}, \boldsymbol{y})$ is optimal for SSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$).

LEMMA 1. *Let* $(\boldsymbol{x}, \boldsymbol{y})$ *be a timed-cycle-free feasible solution for* SSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$)*, and let* $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ *be the solution derived by* (6)–(7)*. It follows that* $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ *is a feasible solution for* USSND($\underline{\mathcal{D}_\mathcal{T}}, \Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$) *with a cost value equal to that of* $(\boldsymbol{x}, \boldsymbol{y})$*. Moreover,* $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ *is an optimal solution to* USSND($\underline{\mathcal{D}_\mathcal{T}}, \Phi_\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$) *provided that* $(\boldsymbol{x}, \boldsymbol{y})$ *is an optimal solution to* SSND($\underline{\mathcal{D}_\mathcal{T}}, \mathcal{K}$).

This problem-relaxation-solution mapping induced by $(\boldsymbol{x}, \boldsymbol{y})$ underlies the feasibility and optimality equivalences between $(\boldsymbol{x}, \boldsymbol{y})$ and $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ established in the next section.

3.3. Feasibility and Optimality Equivalences between (x, y) and $\mathcal{S}(x, y)$

The feasibility of a relaxation solution within a given relaxation model and its optimality relative to the original problem are fundamental to the design of standard DDD algorithms. We introduce three concepts, defined below, to characterize these properties within the DDD framework.

DEFINITION 3. Consider a feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ of SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$).

(i) *Timed feasibility.* The solution $(\boldsymbol{x}, \boldsymbol{y})$ is timed feasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$) if every positive variable in $(\boldsymbol{x}, \boldsymbol{y})$ has a corresponding variable associated with the same timed arc in that model and the candidate solution formed by retaining these positive values and setting all remaining variables in that model to zero satisfies all its constraints.

(ii) *Flat feasibility.* The solution $(\boldsymbol{x}, \boldsymbol{y})$ is flat feasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$) if its flat solution is representable on $\underline{\mathcal{D}_{\mathcal{T}'}}$, meaning that there exists a feasible solution of SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$) whose flat solution is identical to that of $(\boldsymbol{x}, \boldsymbol{y})$.

(iii) *Flat optimality.* The solution $(\boldsymbol{x}, \boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$) if it is optimal for SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) and flat feasible for SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$), meaning that there exists an optimal solution of SCTSNDP($\mathcal{K}$) with the same flat solution and total cost as $(\boldsymbol{x}, \boldsymbol{y})$.

These definitions apply analogously to USCTSNDP($\Phi_{\mathcal{K}}$) by replacing each SSND model with the corresponding USSND model and replacing SCTSNDP($\mathcal{K}$) with USCTSNDP($\Phi_{\mathcal{K}}$).

Throughout the paper, *feasible* retains its standard meaning, indicating that a given solution satisfies all constraints of the corresponding model, whereas *timed feasible* and *flat feasible* refer to the meanings specified in Definition 3. Correspondingly, *timed infeasible* and *flat infeasible* mean not timed feasible and not flat feasible, respectively. The term *feasibility equivalence* refers to equivalence in both timed and flat feasibility, whereas *optimality equivalence* refers specifically to equivalence in flat optimality. The following theorem establishes these equivalences between $(\boldsymbol{x}, \boldsymbol{y})$ and its mapped solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$.

THEOREM 2 **(SSND-USSND Feasibility and Optimality Equivalence).** *Let $(\boldsymbol{x}, \boldsymbol{y})$ be a timed-cycle-free feasible solution to SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) and let $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ denote the mapped solution for the USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) derived by (6)–(7). Then the following statements hold:*

(i) For any valid discretization $\mathcal{T}' \subseteq \hat{\mathcal{T}}$, $(\boldsymbol{x}, \boldsymbol{y})$ is timed feasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$) if and only if $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ is timed feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$). Likewise, $(\boldsymbol{x}, \boldsymbol{y})$ is flat feasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$) if and only if $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ is flat feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$).

(ii) If $(\boldsymbol{x}, \boldsymbol{y})$ is an optimal solution to SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$), then $(\boldsymbol{x}, \boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$) if and only if $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ is flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$).

As noted by Boland et al. (2017), the DDD algorithm for USCTSNDP($\mathcal{K}'$) begins with an initial discretization $\mathcal{T}$ and iteratively solves the relaxation USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}'$). At each iteration, if the optimal relaxation solution is flat optimal, it can be transformed into an optimal solution to the original USCTSNDP($\mathcal{K}'$). Otherwise, the discretization is refined from $\mathcal{T}$ to $\mathcal{T}'$ by adding new time points, thereby making the current relaxation solution timed infeasible or flat infeasible for the subsequent relaxation USSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}'$) and driving the algorithm toward an optimal solution. To this end, Boland et al. (2017) and subsequent studies, such as Marshall et al. (2021) and Shu et al. (2025), propose several well-established algorithmic components for verifying the flat optimality of the current optimal relaxation and for refining the discretization to eliminate the current optimal relaxation solution by making it timed infeasible or flat infeasible in subsequent iterations. By Theorem 2, these established procedures can be applied to $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ to verify the flat optimality of $(\boldsymbol{x}, \boldsymbol{y})$ and, when necessary, refine the discretization to eliminate it from subsequent SSND relaxations. This provides the theoretical foundation for the SDDD algorithm in Section 4.

4. New Exact Algorithm for SCTSNDP

In this section, we present SDDD, a new exact algorithm for the SCTSNDP within the DDD framework that dynamically constructs and exploits solution-induced unsplittable counterparts, thereby avoiding both complete time discretization and a priori demand splitting. Section 4.1 outlines the algorithm, and Sections 4.2–4.5 detail its key components.

4.1. Overview of the SDDD algorithm

The main steps of the SDDD algorithm for SCTSNDP are outlined in Algorithm 1.

The SDDD algorithm begins with an initial discretization $\mathcal{T}$ and a corresponding partially time-expanded network $\underline{\mathcal{D}_{\mathcal{T}}}$ as defined in Section 3.1, where the service arc duration is equal to or shorter than its actual travel time. The initial discretization $\mathcal{T}$ and network $\underline{\mathcal{D}_{\mathcal{T}}}$ are constructed using any established methods from the USCTSNDP literature. As indicated by Theorem 1, this yields an initial relaxation model SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) for SCTSNDP($\mathcal{K}$). In each iteration, we first solve SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) to obtain an optimal relaxation solution $(\boldsymbol{x}, \boldsymbol{y})$, which is necessarily timed-cycle-free, and update the lower bound of the algorithm (LB) based on the optimal objective value of SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$). As indicated in Section 3.2, a mapped USCTSNDP instance, USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$), can be constructed along with its corresponding relaxation over the current discretization $\mathcal{T}$, denoted by USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$). By Lemma 1, a feasible solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) = \mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) with a cost equal to that of $(\boldsymbol{x}, \boldsymbol{y})$ can be derived

Algorithm 1: SDDD Algorithm for the SCTSNDP($\mathcal{K}$)

Input: SCTSNDP defined on a flat network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ with commodity set $\mathcal{K}$; optimality tolerance ε.
Output: Solution $\mathcal{W}$ with cost UB.
begin
1 UB $\leftarrow +\infty$, LB $\leftarrow -\infty$;
2 Initial Relaxation (§4.2): construct an initial discretization $\mathcal{T}$, partially time-expanded network $\underline{\mathcal{D}_{\mathcal{T}}}$, and relaxation model SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$);
3 **while** true **do**
4 Lower Bound Computation (§4.2): Solve SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) to obtain an optimal relaxation solution $(\boldsymbol{x}, \boldsymbol{y})$ with cost $z(\boldsymbol{x}, \boldsymbol{y})$; set $LB \leftarrow \max\{LB, z(\boldsymbol{x}, \boldsymbol{y})\}$;
5 USCTSNDP Counterpart Mapping (§4.3): (i) obtain a feasible split $\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$ of $\mathcal{K}$ induced by $(\boldsymbol{x}, \boldsymbol{y})$; (ii) construct the mapped instance USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$), its relaxation USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$), and a feasible solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ to it;
6 Upper Bound Computation (§4.4): (i) repair $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ into $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$; (ii) derive a feasible solution $\hat{\mathcal{W}}$ to USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) from $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ with cost $z(\hat{\mathcal{W}})$ by using established DDD components from the USCTSNDP literature, which yields a feasible solution to SCTSNDP($\mathcal{K}$) with the same cost;
7 **if** $z(\hat{\mathcal{W}}) < UB$ **then**
8 UB $\leftarrow z(\hat{\mathcal{W}})$, $\mathcal{W} \leftarrow \hat{\mathcal{W}}$;
9 **end**
10 **if** $(UB - LB)/UB \leq \varepsilon$ **then break**;
11 Discretization Refinement (§4.5): refine $\mathcal{T}$ and update $\underline{\mathcal{D}_{\mathcal{T}}}$ based on $S(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$, using established DDD components from the USCTSNDP literature;
12 **end**
13 **return** Solution $\mathcal{W}$ of cost UB;
14 **end**

by (6)–(7). Lemma 1 and Theorem 2 motivate us to utilize the USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) and the solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ to determine whether the algorithm has converged to an optimal solution and, if not, to refine the discretization.

As indicated by Proposition 1, any upper bound for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) also serves as a valid upper bound for the original SCTSNDP($\mathcal{K}$), which can update UB of the algorithm. When the gap between the updated UB and current LB reaches the specified optimality tolerance, the algorithm converges and terminates with an optimal solution for SCTSNDP($\mathcal{K}$) within the optimality tolerance. Otherwise, as part (i) of Theorem 2 implies, discretization $\mathcal{T}$ can be refined to $\mathcal{T}'$ by methods known in the literature on DDD algorithms for the USCTSNDP, so that $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ is no longer timed feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) or, more strongly, the flat solution of $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ is no longer representable in $\underline{\mathcal{D}_{\mathcal{T}'}}$, and thus $(\boldsymbol{x}, \boldsymbol{y})$ is timed infeasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$); in the stronger case, the flat solution of $(\boldsymbol{x}, \boldsymbol{y})$ is no longer representable in $\underline{\mathcal{D}_{\mathcal{T}'}}$. The algorithm then

proceeds to the next iteration with the refined discretization $\mathcal{T}'$. The correctness of Algorithm 1 and its finite convergence to an optimal SCTSNDP solution are established in Section 4.5.

Overall, SDDD avoids both complete time discretization and a priori demand splitting by dynamically mapping SCTSNDP relaxation solutions to solution-induced USCTSNDP counterparts, thereby leveraging established DDD components from the USCTSNDP literature (see, for example, Boland et al. (2017), Marshall et al. (2021), Shu et al. (2025)) to verify optimality and selectively refine the time discretization.

4.2. Initial Relaxation and Lower Bound Computation

Using any valid initialization procedure from the USCTSNDP literature, we construct an initial discretization $\mathcal{T}$ and network $\underline{\mathcal{D}_{\mathcal{T}}}$, which yield a valid initial relaxation SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) of SCTSNDP($\mathcal{K}$) by Theorem 1. We further strengthen this relaxation with the following valid inequalities adapted from the USCTSNDP.

$$\sum_{((i,t),(j,\bar{t}))\in\underline{\mathcal{A}_{\mathcal{T}}}} \tau_{ij} x_{ij}^{kt\bar{t}} \le l^k - e^k, \quad \forall k \in \mathcal{K}, \tag{8}$$

$$\max\left\{1, \frac{q^k}{u_{ij}}\right\} x_{ij}^{kt\bar{t}} \le y_{ij}^{t\bar{t}}, \quad \forall\ ((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}, k \in \mathcal{K}. \tag{9}$$

Inequalities (8) bound the flow-weighted average travel time across all paths used by each commodity $k \in \mathcal{K}$. Inequalities (9) ensure that any used service arc has an associated vehicle dispatch with sufficient capacity for the transported fraction of commodity $k \in \mathcal{K}$. Details of how these inequalities are adapted from the USCTSNDP literature, together with proofs of their validity for the SCTSNDP, are provided in Appendix EC.5. Therefore, at each iteration of the SDDD, we solve the strengthened relaxation model defined by (1)–(5), together with (8) and (9), to obtain a lower bound on SCTSNDP($\mathcal{K}$). The notions of flat feasibility and flat optimality in Definition 3 apply unchanged to this strengthened relaxation. To preserve the feasibility and optimality equivalences established in Theorem 2, the mapped USSND relaxation is strengthened with a corresponding inequality expressed in terms of the sub-commodities, as described in Section 4.3.

Additionally, some other techniques tightening relaxation, which are originally proposed for the USCTSNDP, can also be adopted for the SCTSNDP. In particular, the significant time points proposed by Shu et al. (2025), which eliminate infeasible consolidations arising from incompatible time windows, remain valid for splittable deliveries because all flow splits of a commodity share its time window and can therefore be incorporated into the initial discretization.

4.3. USCTSNDP Counterpart Mapping

Given the optimal solution $(\boldsymbol{x}, \boldsymbol{y})$ of SSND$(\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K})$, this section details the demand-splitting scheme used to generate the feasible split $\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$ of the original commodity set $\mathcal{K}$, and the construction of the solution-induced USCTSNDP counterparts introduced in Section 3.2.

We generate a split $\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$ using a flow-decomposition procedure adapted from Ahuja et al. (1993). Specifically, for each commodity $k \in \mathcal{K}$, let $\boldsymbol{x}^k = \{x_{ij}^{kt\bar{t}}\}_{((i,t),(j,\bar{t}))\in \underline{\mathcal{A}_{\mathcal{T}}} \cup \underline{\mathcal{H}_{\mathcal{T}}}}$ denote its continuous arc-flow fractions under $(\boldsymbol{x}, \boldsymbol{y})$. Initialize the residual flow as $\boldsymbol{x}^{k,\text{res}} = \boldsymbol{x}^k$. The flow-decomposition procedure then repeatedly extracts an origin-to-destination timed path $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ using only arcs with positive residual flow and set $\alpha^{k,m} = \min_{a\in P^{k,m}(\boldsymbol{x},\boldsymbol{y})} x_a^{k,\text{res}}$. The corresponding flow amount assigned to the extracted path is $g^{k,m}(\boldsymbol{x}, \boldsymbol{y}) = q^k \alpha^{k,m}$. We then subtract $\alpha^{k,m}$ from the residual flow on every arc of the path and repeat until no positive residual flow remains. This procedure yields the flow splits $M_k(\boldsymbol{x}, \boldsymbol{y})$ and their timed paths $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ and flow amounts $g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, satisfying $\sum_{m\in M_k(\boldsymbol{x},\boldsymbol{y})} g^{k,m}(\boldsymbol{x}, \boldsymbol{y}) = q^k$. Accordingly, the set $\Gamma_k(\boldsymbol{x}, \boldsymbol{y}) = \{\gamma_m^k\}_{m\in M_k(\boldsymbol{x},\boldsymbol{y})}$, where each sub-commodity γ_m^k is a copy of k and has demand $q^{\gamma_m^k} = g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, constitutes a feasible demand disaggregation for commodity $k \in \mathcal{K}$. Aggregating these sets across all commodities yields the feasible split $\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y}) = \bigcup_{k\in\mathcal{K}} \Gamma_k(\boldsymbol{x}, \boldsymbol{y})$ of $\mathcal{K}$.

Based on the generated $\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$, we construct the mapped instance USCTSNDP$(\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y}))$ and its relaxation USSND$(\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y}))$. The latter is the binary-flow time-indexed formulation introduced in Section 2.3, strengthened by the following counterpart of (8):

$$\sum_{\gamma_m^k\in\Gamma_k(\boldsymbol{x},\boldsymbol{y})} \frac{q^{\gamma_m^k}}{q^k} \sum_{((i,t),(j,\bar{t}))\in\underline{\mathcal{A}_{\mathcal{T}}}} \tau_{ij} x_{ij}^{\gamma_m^k t\bar{t}} \le l^k - e^k, \quad \forall k \in \mathcal{K}. \tag{10}$$

The mapped solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is then derived from (6)–(7). By construction, $x_{ij}^{kt\bar{t}} = \sum_{\gamma_m^k\in\Gamma_k(\boldsymbol{x},\boldsymbol{y})} (q^{\gamma_m^k}/q^k) \bar{x}_{ij}^{\gamma_m^k t\bar{t}}$. Hence, (8) and (10) have identical left-hand sides under the mapping. The same identity applies when a feasible mapped USSND solution is aggregated by original commodity. Therefore, the two inequalities are preserved in both directions, and Theorem 2 continues to hold for the strengthened relaxations. No separate counterpart of (9) is needed to preserve this correspondence because this inequality, as well as its binary-flow analogue, is implied by the capacity constraints and the integrality of the vehicle variables. When the flow decomposition is not unique, the mapped counterparts are defined with respect to any fixed valid decomposition.

4.4. Upper Bound Computation

Consider an optimal solution $(\boldsymbol{x}, \boldsymbol{y})$ of SSND$(\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K})$. By Proposition 1, any feasible solution of the mapped USCTSNDP instance USCTSNDP$(\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y}))$ provides an upper bound for the

SCTSNDP($\mathcal{K}$). This motivates us to utilize a feasible solution of USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) to update the UB within the SDDD algorithm.

Within the DDD algorithms for the USCTSNDP, various well-established upper-bound derivation methods exist to derive a feasible solution for the USCTSNDP from a feasible solution of its relaxation, provided each commodity k follows a k-feasible flat path. In SDDD, however, even the strengthen relaxation formulation with inequalities (8) do not guarantee that every flat path followed by a flow split of commodity k in $(\boldsymbol{x},\boldsymbol{y})$, consequently by the corresponding sub-commodity in $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$, is k-feasible. This prevents the direct application of standard upper-bound derivation methods from the USCTSNDP literature to $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ for obtaining a feasible solution to USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$), and thus a valid upper bound for the original SCTSNDP($\mathcal{K}$), at every iteration of the SDDD algorithm.

Proposition 3 below provides the theoretical basis for overcoming this issue. Guided by this proposition, we develop a solution-repair procedure that transforms $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ into $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$, which is feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) with each commodity $k \in \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$ following a k-feasible flat path, thereby allowing standard upper-bound derivation methods from the USCTSNDP literature to be applied to $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$ to update the UB at every iteration of SDDD.

PROPOSITION 3. *Let $(\boldsymbol{x},\boldsymbol{y})$ be an optimal solution of SSND($\underline{\mathcal{D}_{\mathcal{T}}},\mathcal{K}$) satisfying constraints (8). For each commodity $k \in \mathcal{K}$, let $M_k(\boldsymbol{x},\boldsymbol{y})$ be its set of flow splits and let $P^{k,m}(\boldsymbol{x},\boldsymbol{y})$ be the associated timed-path over $\underline{\mathcal{D}_{\mathcal{T}}}$ for each split $m \in M_k(\boldsymbol{x},\boldsymbol{y})$, both generated as described in Section 4.3. Then, there exists at least one flow split $m \in M_k(\boldsymbol{x},\boldsymbol{y})$ such that its timed-path $P^{k,m}(\boldsymbol{x},\boldsymbol{y})$ corresponds to a k-feasible flat path.*

We now detail the solution-repair procedure. Consider the optimal solution $(\boldsymbol{x},\boldsymbol{y})$ of SSND($\underline{\mathcal{D}_{\mathcal{T}}},\mathcal{K}$), along with the flow splits $\{M_k(\boldsymbol{x},\boldsymbol{y})\}_{k\in\mathcal{K}}$ and the corresponding timed-paths $\{P^{k,m}(\boldsymbol{x},\boldsymbol{y})\}$ and flow amounts $\{g^{k,m}(\boldsymbol{x},\boldsymbol{y})\}$ for each $m \in M_k(\boldsymbol{x},\boldsymbol{y})$ as generated in Section 4.3. By Proposition 3, for each commodity $k \in \mathcal{K}$, select a split $\hat{m}_k \in M_k(\boldsymbol{x},\boldsymbol{y})$ whose timed path corresponds to a k-feasible flat path. For every $m \in M_k(\boldsymbol{x},\boldsymbol{y})$, retain $P^{k,m}(\boldsymbol{x},\boldsymbol{y})$ if it corresponds to a k-feasible flat path and otherwise replace it with $P^{k,\hat{m}_k}(\boldsymbol{x},\boldsymbol{y})$. Denote the resulting path by $\bar{P}^{k,m}(\boldsymbol{x},\boldsymbol{y})$. The repair preserves the flow amount $g^{k,m}(\boldsymbol{x},\boldsymbol{y})$ of each split and changes only its timed path when necessary. The repaired solution $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = (\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ is then defined by (11)–(12).

$$\hat{\boldsymbol{x}}_{ij}^{\gamma_m^k t\bar{t}} = \begin{cases} 1, & \text{if } ((i,t),(j,\bar{t})) \in \bar{P}^{k,m}(\boldsymbol{x},\boldsymbol{y}), \\ 0, & \text{otherwise,} \end{cases} \qquad \forall k \in \mathcal{K}, m \in M_k(\boldsymbol{x},\boldsymbol{y}), ((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}} \cup \underline{\mathcal{H}_{\mathcal{T}}} \tag{11}$$

$$\hat{\boldsymbol{y}}_{ij}^{t\bar{t}} = \left\lceil \frac{\sum_{k\in\mathcal{K}} \sum_{m\in M_k(\boldsymbol{x},\boldsymbol{y})} g^{k,m}(\boldsymbol{x},\boldsymbol{y}) \cdot \hat{\boldsymbol{x}}_{ij}^{\gamma_m^k t\bar{t}}}{u_{ij}} \right\rceil, \quad \forall((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}. \tag{12}$$

By Proposition 4, $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$ constitutes an alternative feasible solution to USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$).

PROPOSITION 4. *Let $(\boldsymbol{x},\boldsymbol{y})$ be an optimal solution to SSND($\underline{\mathcal{D}_{\mathcal{T}}},\mathcal{K}$) satisfying constraints (8) and (9), and let $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$ be the repaired solution derived from Equations (11) and (12). Then: (i) $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$ is feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$), and each sub-commodity follows a k-feasible flat path; (ii) if $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for the SCTSNDP($\mathcal{K}$), then $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{S}(\boldsymbol{x},\boldsymbol{y})$.*

Subsequently, any standard upper-bound derivation methods for USCTSNDP can be applied to the repaired solution $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$ to obtain a feasible solution for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$), thereby updating the UB of the SDDD algorithm at every iteration.

We next show that the upper-bound procedure described above yields $UB = LB$ whenever $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$). Suppose that $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal with cost LB. By Lemma 1 and Theorem 2(ii), the mapped solution $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ is an optimal solution to USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) with cost LB and is flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$). Proposition 4(ii) then implies that the repaired solution $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$ is identical to $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ and hence is also flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) with cost LB. When applied to the flat-optimal solution $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y})$, standard upper-bound derivation methods for the USCTSNDP return a feasible solution to USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) with the same cost LB. Updating the upper bound then yields $UB = LB$. Hence, whenever the relaxation solution $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$), the SDDD algorithm is guaranteed to attain $UB = LB$ and terminate with an optimal solution to SCTSNDP($\mathcal{K}$).

4.5. Discretization Refinement

If $UB > LB$, Theorem 2, together with the upper-bound procedure detailed in Section 4.4, implies that neither the relaxation solution $(\boldsymbol{x},\boldsymbol{y})$ nor its mapped solution $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$) or USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$), respectively. In this case, we must refine the discretization $\mathcal{T}$ to $\mathcal{T}'$ so that $(\boldsymbol{x},\boldsymbol{y})$ becomes timed infeasible for the refined model SSND($\underline{\mathcal{D}_{\mathcal{T}'}},\mathcal{K}$) or, more strongly, its flat solution can no longer be represented on $\underline{\mathcal{D}_{\mathcal{T}'}}$. As indicated by Theorem 2, this is equivalent to refining the discretization using established procedures from the USCTSNDP literature, either to render the mapped solution $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ timed infeasible for USSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) (Boland et al. 2017, Shu et al. 2024) or make its flat solution unrepresentable in $\underline{\mathcal{D}_{\mathcal{T}'}}$ (Marshall et al. 2021, Shu et al. 2025). In SDDD, any of these methods can be applied directly to the mapped solution $S(\boldsymbol{x},\boldsymbol{y})$ to refine the current discretization $\mathcal{T}$ by incorporating new time points, and at

least one new time point, triggered from the solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$, must be added to the discretization at each iteration where refinement is required. Therefore, with an optimality tolerance of zero, the discretization is guaranteed to converge to the complete discretization within a finite number of iterations. This ultimately ensures the convergence of the algorithm to an optimal SCTSNDP($\mathcal{K}$) solution in a finite number of iterations, thereby establishing the following Theorem 3.

THEOREM 3. *With an optimality tolerance of zero, the SDDD algorithm terminates after finitely many iterations with an optimal solution to SCTSNDP*($\mathcal{K}$).

5. Computational Study

In this section, we conduct a comprehensive computational study to evaluate the efficiency and effectiveness of the SDDD algorithm described in Algorithm 1. Section 5.1 first demonstrates the challenges caused by time discretization and demand splitting, thus providing motivation for the SDDD. We then assess the effectiveness and computational efficiency of the SDDD in Section 5.2. Furthermore, we investigate the impact of splittable deliveries on cost savings in Section 5.3 , first on the standard benchmark instances and then on newly generated instances.

We use two set of instances in the following computational study. The first one consists of 558 standard CTSNDP instances studied in Boland et al. (2017), Marshall et al. (2021), Shu et al. (2025), referred to benchmark instances. These instances are used in Section 5.1-5.2 to evaluate computational performance and in Section 5.3 to examine the cost benefits of splittable deliveries. These instances are classified by time flexibility and cost ratio. An instance is classified as low time flexibility (LF) if $\min_{k\in\mathcal{K}}\{l^k - (e^k + \phi^k(o^k, d^k))\} < 227$, where $\phi^k(o^k, d^k)$ denotes the shortest-path travel time of commodity k, and as high time flexibility (HF) otherwise, and as low cost ratio (LC) if $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\frac{f_a}{c_a u_a} < 0.175$ and as high cost ratio (HC) otherwise, where $c_a = c_a^k$ for all $k \in \mathcal{K}$, yielding four groups: HC/LF, HC/HF, LC/LF, and LC/HF. The second one consists of 252 newly generated instances introduced in Section 5.3.2, which are used to further investigate the cost benefits under different parameter settings. All experiments were conducted on a PC with a 3.2 GHz CPU and 64 GB of RAM. Each run was terminated once either the optimality gap dropped below 1% or the time limit of 1 hour was reached. All computational methods were implemented in Java, and Gurobi 10.0.2 was used to solve the resulting MIP formulations.

5.1. Impact of Time Discretization and Demand Splitting on Model Sizes

In this section, we examine how the granularity of time discretization and demand splitting affects formulation size across all baseline instances, where demand-splitting granularity ϵ partitions each commodity k into $\lfloor q^k/\epsilon \rfloor$ sub-commodities of demand ϵ, plus one for any remainder.

Figure 1 **Impact of time discretization on SSND($\mathcal{D}^{\Delta}_{\mathcal{T}}, \mathcal{K}$) model size.**

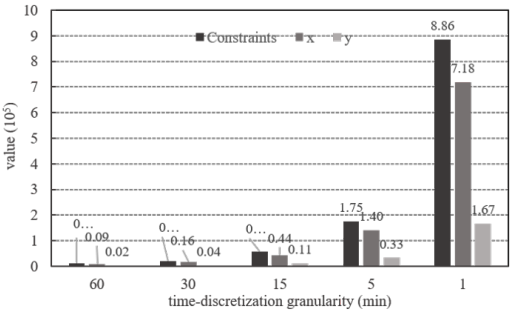


Figure 2 **Impact of demand splitting on commodity set $\Phi_{\mathcal{K}}$ size.**

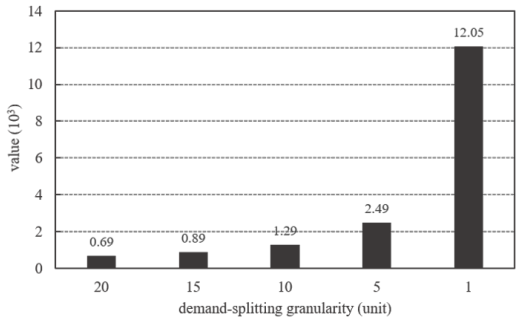


Figure 1 shows the average numbers of variables and constraints in the continuous-flow time-indexed model SSND($\mathcal{D}^{\Delta}_{\mathcal{T}}, \mathcal{K}$) increase dramatically, as the time discretization unit Δ decreases from 60 minutes to 1 minute. In particular, setting $\Delta = 1$ leads to a prohibitively large mixed-integer program, making it computationally challenging to obtain an optimal solution to the original SCTSNDP by directly solving this complete model.

Figure 2 shows the size of feasible split $\Phi_{\mathcal{K}}$, which is measured by the total number of sub-commodities, grows rapidly as the demand-splitting granularity decreases from 20 to 1. As mentioned in Section 2.3, smaller demand-splitting granularity provides more flexibility in representing flow-splitting decisions. However, it also substantially enlarges the commodity set and, consequently, the binary-flow time-indexed formulations over partially time-expanded networks solved by standard DDD algorithms, which may significantly reduce their computational efficiency.

Overall, Figures 1 and 2 illustrate the growth in computational scale of the two approaches discussed in Section 2.4, whose computational performance is examined in the next section.

5.2. Performance of the SDDD Algorithm

In this section, we evaluate the proposed SDDD against the two approaches introduced in Section 2.4. Specifically, we compare the following three methods: (1) SDDD, which implements the general framework in Algorithm 1 using the DDD components of Shu et al. (2025) for the mapped USCTSNDP counterparts and the adaptive optimality-tolerance strategy of Marshall et al. (2021); (2) SMIP, an exact method that directly solves SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$) using a commercial solver; and (3) USDDD$_1$, a heuristic method for the SCTSNDP that applies the DDD algorithm of Shu et al. (2025), with the same adaptive optimality-tolerance strategy as SDDD, to USCTSNDP($\Phi^1_{\mathcal{K}}$), where every sub-commodity in $\Phi^1_{\mathcal{K}}$ has a demand of 1, which is possible because all instances have integer-valued q^k. We note that both SDDD and USDDD$_1$ adopt the solution-bundle strategy of Shu et al. (2025), which processes the current optimal relaxation solution and selected

additional feasible solutions at each iteration to generate upper-bound candidates and refinement points. To apply the solution-bundle and adaptive optimality-tolerance strategies within SDDD, we remove any positive-flow directed cycles from each obtained solution before mapping it, without affecting feasibility or increasing cost. The resulting timed-cycle-free solution is then mapped and repaired using the procedures in Sections 4.3 and 4.4. Because the current optimal relaxation solution remains the basis for the lower bound and termination test, incorporating auxiliary solutions does not affect the exactness of SDDD. Moreover, for a fair comparison, the MIP/IP solver is restricted to a single thread for all methods.

Table 1 Comparison of Methods for SCTSNDP

	SDDD				SMIP				$USDDD_1$			
	G%	T(s)	Opt%	Fes%	G%	T(s)	Opt%	Fes%	G%	T(s)	Opt%	Fes%
HC/HF(177)	0.85	153.77	100.00	100.00	46.65	2114.02	49.72	59.89	51.64	3518.79	5.08	72.88
HC/LF(183)	0.74	10.51	100.00	100.00	4.81	458.87	94.54	95.63	28.94	2788.74	26.23	83.61
LC/HF(104)	0.31	0.20	100.00	100.00	0.26	58.41	99.04	100.00	21.63	2763.20	67.31	78.85
LC/LF(94)	0.42	0.41	100.00	100.00	0.27	4.11	100.00	100.00	2.51	729.74	95.74	97.87

Table 1 reports the computational results for three methods solving 558 standard CTSNDP instances, including the average optimality gap (G%), i.e. $100.0 \times \frac{(UB-LB)}{UB}$, the average computing time in seconds (T(s)), the percentage of instances solved within the 1% optimality tolerance (Opt%), and the percentage of instances for which a feasible solution is found (Fes%) within the time limit. We note that for $USDDD_1$, G% and Opt% are evaluated with respect to the corresponding USCTSNDP restriction. The results show that SDDD achieves the best overall performance in terms of both solution quality and computational efficiency. It solves all instances across all groups to the prescribed 1% optimality tolerance within one hour. By contrast, due to memory limitations, SMIP and $USDDD_1$ fail to find any feasible solution for around 14% and 18% of the instances, respectively. For these failed instances, the optimality gap and runtime are set to 100% and 3600 seconds, respectively, when calculating the averages reported in Table 1. The advantage of SDDD is particularly pronounced in the HC groups. For HC/HF and HC/LF instances, it achieves average gaps of only 0.85% and 0.74% in 153.77 and 10.51 seconds, respectively, compared with corresponding gaps of 46.65% and 4.81% for SMIP and 51.64% and 28.94% for $USDDD_1$. These results demonstrate the effectiveness of SDDD across a broad range of benchmark instances.

5.3. Cost Benefits from Splittable Deliveries

In this section, we investigate the cost benefits of allowing splittable deliveries in the CTSNDP. We use SDDD, implemented as described in Section 5.2 to solve each considered SCTSNDP instance,

and apply EDDD proposed by Shu et al. (2025) to the corresponding USCTSNDP restriction, denoted as USDDD. The observed cost saving of each instance is measured as $\frac{Z_{US}-Z_S}{Z_{US}} \times 100\%$, where Z_{US} and Z_S are the best feasible solution costs returned by USDDD and SDDD, respectively. We first report the cost savings on the benchmark instances, and then conduct experiments on newly generated instances to examine how these savings vary across parameter settings.

5.3.1. Results on Benchmark Instances The results on the 558 benchmark instances show only limited observed cost benefits from splittable deliveries, while SDDD and USDDD require comparable computational effort (see EC.3). Only about 44.4% of the instances yield a lower-cost solution under SDDD than under USDDD, and the observed cost savings are small, typically around 0.2%, with the max saving almost 1.0%.

The limited cost savings observed here align with the findings of Archetti et al. (2008), who show that splittable deliveries provide limited benefits when demands are small relative to vehicle capacity. Indeed, the average arc capacity in our benchmark instances is approximately 27 times the average commodity demand, placing them in exactly such a setting. This observation motivates the broader investigation in the next subsection, which examines the cost benefits of splittable deliveries across instances with more diverse demand-to-capacity settings.

5.3.2. Results on Newly Generated Instances This section examines the cost benefits of splittable deliveries under different levels of demand-to-capacity settings, along with the associated solution structures underlying these benefits. Motivated by previous studies showing that tighter capacities increase the value of split deliveries (Desaulniers 2010, Archetti et al. 2008), we construct a new test set by adapting the 126 timed benchmark instances with at most 40 commodities used in Section 5.2 by setting every physical-arc capacity to either 1.0 or 1.5 times the average commodity demand, compared with at least 2.5 times in the original instances. These variants are referred to as tight-capacity (TC) and moderate-capacity (MC) instances, respectively. Moreover, we set the variable flow costs to zero while retaining the original fixed dispatch costs, allowing us to more clearly assess the cost savings resulting from changes in vehicle-dispatch plans enabled by splittable deliveries. This yields a final test set of $126 \times 2 = 252$ instances, classified into four groups, TC/HF, TC/LF, MC/HF, and MC/LF, based on the TC/MC capacity settings and the HF/LF time-flexibility classification introduced at the beginning of Section 5.

Among all 252 newly generated instances, 98.41% yield a lower-cost solution under SDDD than under USDDD, with an average observed saving of about 2.7% and a maximum of 7.32%. These

results indicate that allowing commodity splitting can generate meaningful cost savings in some instances and therefore should not be overlooked in CTSNDP. To further examine how splittable deliveries affect solution structure, Table 2 reports detailed results for the four instance groups. We represent each consolidation by a triple comprising a timed arc, a set of at least two distinct original commodities traversing that arc, and their corresponding flow quantities. Let C_S and C_{US} denote the resulting sets of consolidations in the SCTSNDP and USCTSNDP solutions, respectively, and let $N_S = |C_S|$ and $N_{US} = |C_{US}|$. In addition to the cost saving, Table 2 reports three measures of structural differences: (i) the change in the number of consolidations, $\delta_{N_c} = N_S - N_{US}$; (ii) the consolidation similarity, SC = $\frac{|C_S \cap C_{US}|}{|C_S \cup C_{US}|} \times 100\%$; and (iii) the percentage of split commodities, SP = $\frac{|\mathcal{K}_S|}{|\mathcal{K}|} \times 100\%$, where $\mathcal{K}_S$ is the set of original commodities split into at least two flows in the SCTSNDP solution. For comparability, we include only instances for which both SDDD and USDDD achieve the prescribed optimality tolerance, and report group averages.

Table 2 Summary Results on the Cost-saving Instances

	Cost Saving						Solution Structure		
Group	ave	max	$\leq 1\%$	$(1\%, 2\%]$	$(2\%, 3\%]$	> 3	δ_{N_c}	SC	SP
MC/LF(19)	1.00	3.36	52.63	31.58	10.53	5.26	5.05	36.16	8.16
MC/HF(75)	2.10	5.99	34.67	17.33	17.33	30.67	7.51	26.18	11.15
TC/LF(19)	2.01	5.12	31.58	31.58	15.79	21.05	5.79	30.80	18.82
TC/HF(75)	2.96	7.32	13.33	25.33	17.33	44.00	8.07	26.80	22.95

Table 2 shows that cost benefits from splittable deliveries become larger under tighter capacity and greater temporal flexibility. The average cost savings increase from 1.88% under MC to 2.77% under TC and from 1.51% under LF to 2.53% under HF. The distribution of cost savings provides further evidence of these differences across the settings. The proportion of instances achieving savings above 3% increases from 5.26% under MC/LF to 44.00% under TC/HF. Thus, larger savings are more prevalent when tight capacity and high temporal flexibility occur together. These cost savings are also associated with notable changes in solution structure. From MC/LF to TC/HF, the average SP increases from 8.16% to 22.95%, δ_{N_c} increases from 5.05 to 8.07, and SC decreases from 36.16% to 26.80%. Overall, larger cost savings are accompanied by more frequent commodity splitting and a larger number of consolidations.

Although the newly generated instances demonstrate that commodity splitting can yield meaningful cost savings under certain conditions, splitting may also incur additional operational costs, such as holding and hub-transfer costs. Appendix EC.4 discusses how our algorithm framework can be extended to incorporate such costs. In practice, whether to split a commodity depends on

a comprehensive assessment of all relevant operational costs, and our algorithmic framework provides a foundation for incorporating these considerations.

6. Conclusions

In this paper, we propose an exact solution method for the SCTSNDP that neither requires solving the continuous-flow formulation on the complete time-expanded network nor relies on splitting commodities in advance. The method extends the DDD framework for the USCTSNDP by constructing solution-dependent commodity splits during the solution process. Specifically, in each iteration of the algorithm, we construct an associated USCTSNDP instance, its relaxation model, and a mapped solution, from the corresponding relaxation solution of the SCTSNDP. The main theoretical contribution is the establishment of feasibility and optimality equivalences between each SCTSNDP relaxation solution and its mapped USCTSNDP counterpart. Based on these equivalences, the algorithm iteratively solves the SCTSNDP relaxation on a partially time-expanded network to obtain a lower bound and leverages the associated USCTSNDP counterparts to derive valid upper bounds and refine the time discretization until optimality is certified. Computational results show that the proposed method outperforms the existing methods for the SCTSNDP. Additional experiments show that splitting yields greater cost savings under tight capacity and flexible commodity time windows. Additional analysis also demonstrates that the proposed algorithmic framework can be extended to incorporate additional operational costs associated with splitting. In future research, this framework may be adapted to other transportation optimization problems involving splittable deliveries, such as split-delivery vehicle routing problems with time windows.

References

Ahuja RK, Magnantl TL, Orlin JB (1993) *Network flows: theory, algorithms, and applications* (Prentice-hall).

Archetti C, Campbell AM, Speranza MG (2016) Multicommodity vs. single-commodity routing. *Transportation Science* 50(2):461–472.

Archetti C, Savelsbergh MW, Speranza MG (2006) Worst-case analysis for split delivery vehicle routing problems. *Transportation science* 40(2):226–234.

Archetti C, Savelsbergh MW, Speranza MG (2008) To split or not to split: That is the question. *Transportation Research Part E: Logistics and Transportation Review* 44(1):114–123.

Atamtürk A, Rajan D (2002) On splittable and unsplittable flow capacitated network design arc–set polyhedra. *Mathematical Programming* 92(2):315–333.

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

Crainic TG (2000) Service network design in freight transportation. *European Journal of Operational Research* 122(2):272–288.

Crainic TG, Hewitt M (2021) Service network design. Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 347–382 (Springer).

Crainic TG, Hewitt M, Toulouse M, Vu DM (2016) Service network design with resource constraints. *Transportation Science* 50(4):1380–1393.

Desaulniers G (2010) Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research* 58(1):179–192.

Dror M, Trudeau P (1989) Savings by split delivery routing. *Transportation Science* 23(2):141–145.

Gouveia L, Leitner M, Ruthmair M (2023) Multi-depot routing with split deliveries: Models and a branch-and-cut algorithm. *Transportation Science* 57(2):512–530.

He EY, Boland N, Nemhauser G, Savelsbergh M (2022) Dynamic discretization discovery algorithms for time-dependent shortest path problems. *INFORMS Journal on Computing* 34(2):1086–1114.

He Y, Hewitt M, Lehuédé F, Medina J, Péton O (2024) A continuous-time service network design and vehicle routing problem. *Networks* 83(2):300–323.

Hewitt M (2019) Enhanced dynamic discretization discovery for the continuous time load plan design problem. *Transportation Science* 53(6):1731–1750.

Hewitt M (2022) The flexible scheduled service network design problem. *Transportation Science* 56(4):1000–1021.

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

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

Ocampo-Giraldo DM, Anaya-Arenas AM, Contardo C (2025) An iterative exact algorithm over a time-expanded network for the transportation of biomedical samples. *INFORMS Journal on Computing* 37(5):1242–1266.

Sherali HD, Smith JC, Lee Y (2000) Enhanced model representations for an intra-ring synchronous optical network design problem allowing demand splitting. *INFORMS Journal on Computing* 12(4):284–298.

Shu S, Xu Z, Baldacci R (2024) Incorporating holding costs in continuous-time service network design: New model, relaxation, and exact algorithm. *Transportation Science* 58(2):412–433.

Shu S, Xu Z, Baldacci R (2025) New dynamic discretization discovery strategies for continuous-time service network design, 10.2139/ssrn.5099677.

Van Dyk M, Koenemann J (2024) Sparse dynamic discretization discovery via arc-dependent time discretizations. *Computers & Operations Research* 169:106715.

van Lieshout RN, van der Schaft T (2026) Dynamic discretization discovery for the multidepot vehicle scheduling problem with trip shifting. *INFORMS Journal on Computing* 38(3):958–981.

Online Supplement: A New Exact Algorithm for Continuous-Time Service Network Design with Splittable Deliveries

EC.1. Notation and Structural Relationships

For ease of reference, Figure EC.1 illustrates the relationships among the problem variants, their time-indexed formulations, and the associated partial-network relaxations.

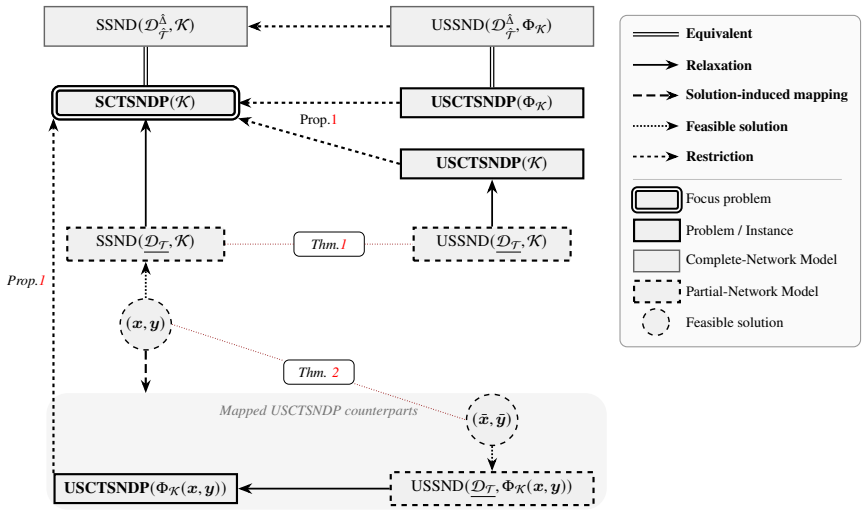


Figure EC.1 Relationships among problem variants, formulations, and relaxations.

EC.2. Proof of Statements

EC.2.1. Proof of Proposition 1

Proof Let $\Phi_{\mathcal{K}} = \bigcup_{k\in\mathcal{K}} \Gamma_k$ be a feasible split of $\mathcal{K}$. For each original commodity $k \in \mathcal{K}$, each sub-commodity $\gamma \in \Gamma_k$ inherits the origin, destination, earliest available time, due time, and unit flow cost of k, and satisfies $\sum_{\gamma\in\Gamma_k} q^\gamma = q^k$. Consider any feasible solution to USCTSNDP($\Phi_{\mathcal{K}}$), each sub-commodity $\gamma \in \Gamma_k$ is transported along a k-feasible flat path $p^\gamma = (a^\gamma_1, \ldots, a^\gamma_{|p^\gamma|})$ with associated dispatch times along the delivery path $t^\gamma = (t^\gamma_{a^\gamma_1}, \ldots, t^\gamma_{a^\gamma_{|p^\gamma|}})$, where $t^\gamma_{a^\gamma_1} \geq e^k, t^\gamma_{a^\gamma_n} \geq t^\gamma_{a^\gamma_{n-1}} + \tau_{a^\gamma_{n-1}}$ for each $n = 2, \ldots, |p^\gamma|$, and $t^\gamma_{a^\gamma_{|p^\gamma|}} + \tau_{a^\gamma_{|p^\gamma|}} \leq l^k$. Therefore, a feasible solution of USCTSNDP($\Phi_{\mathcal{K}}$) can be represented as $\{(p^\gamma, t^\gamma)\}_{\gamma\in\Gamma_k, k\in\mathcal{K}}$.

We now construct a feasible solution to SCTSNDP($\mathcal{K}$) from the solution $\{(p^\gamma, t^\gamma)\}_{\gamma\in\Gamma_k, k\in\mathcal{K}}$ while preserving the same cost. For each commodity $k \in \mathcal{K}$, create one flow split corresponding to each sub-commodity $\gamma \in \Gamma_k$. For this flow split, define the triple $(q^\gamma, p^\gamma, t^\gamma)$, where q^γ is its flow amount, p^γ is its delivery path, and t^γ gives its dispatch times. In particular, q^γ units of commodity k are transported along the same delivery path and with the same dispatch times as sub-commodity γ in the feasible USCTSNDP($\Phi_{\mathcal{K}}$) solution $\{(p^\gamma, t^\gamma)\}_{\gamma\in\Gamma_k, k\in\mathcal{K}}$. The resulting

solution $\{(q^\gamma, p^\gamma, t^\gamma)\}_{\gamma\in\Gamma_k,\, k\in\mathcal{K}}$ is therefore feasible for SCTSNDP($\mathcal{K}$). The vehicle dispatches and the associated consolidation plan remain unchanged, except sub-commodities are relabeled as flow splits of their original commodities. Hence, the two solutions have the same objective value.

Therefore, every feasible solution to USCTSNDP($\Phi_\mathcal{K}$) corresponds to a feasible solution to SCTSNDP($\mathcal{K}$) with the same objective value, which proves Proposition 1. □

EC.2.2. Proof of Corollary 1

Proof Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an optimal solution of SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$) with cost $z(\boldsymbol{x}^*, \boldsymbol{y}^*)$. Since the complete time-expanded network is finite and acyclic, the feasible flow $q^k\boldsymbol{x}^{k*}$ of each commodity $k \in \mathcal{K}$ admits a finite decomposition into source-to-destination timed paths, resulting flow splits $M_k(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for each commodity $k \in \mathcal{K}$ with associated delivery quantities $g^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*)$, delivery paths $p^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*)$ and dispatch schedules $t^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*)$. For each flow split $m \in M_k(\boldsymbol{x}^*, \boldsymbol{y}^*)$, create a sub-commodity $\gamma^k_m \in \Gamma_k(\boldsymbol{x}^*, \boldsymbol{y}^*)$ with demand $q^{\gamma^k_m} = g^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*)$ as specified in Definition 1. Since the split quantities are positive and sum to q^k, a finite feasible split $\bar{\Phi}_\mathcal{K} = \Phi_\mathcal{K}(\boldsymbol{x}^*, \boldsymbol{y}^*) = \cup_{k\in\mathcal{K}}\Gamma_k(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is defined following the definition 2.

We now construct a feasible solution to USCTSNDP($\bar{\Phi}_\mathcal{K}$) from the solution $\{(g^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*), p^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*), t^{k,m}(\boldsymbol{x}^*, \boldsymbol{y}^*)\}_{m\in M_k(\boldsymbol{x}^*,\boldsymbol{y}^*),\, k\in\mathcal{K}}$ while preserving the same cost. For each sub-commodity $\gamma^k_m \in \Gamma_k(\boldsymbol{x}^*, \boldsymbol{y}^*)$, create one flow $\{(p^{\gamma^k_m}, t^{\gamma^k_m})\}$. In particular, the sub commodity is transported along the same delivery path and with the same dispatch times as in the demand splitting in the $(\boldsymbol{x}^*, \boldsymbol{y}^*)$. Therefore, the resulting solution is feasible for USCTSNDP($\bar{\Phi}_\mathcal{K}$). The vehicle dispatches and the associated consolidation plan remain unchanged. Hence, $\{(p^{\gamma^k_m}, t^{\gamma^k_m})\}_{m\in M_k(\boldsymbol{x}^*,\boldsymbol{y}^*),\, k\in\mathcal{K}}$ has the cost $z(\boldsymbol{x}^*, \boldsymbol{y}^*)$. By Proposition 1, any feasible solution to USCTSNDP($\bar{\Phi}_\mathcal{K}$) with cost less than $z(\boldsymbol{x}^*, \boldsymbol{y}^*)$ would yield a feasible solution to SCTSNDP($\mathcal{K}$) with the same lower cost. Since $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ has the lowest cost for SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$), $\{(p^{\gamma^k_m}, t^{\gamma^k_m})\}_{m\in M_k(\boldsymbol{x}^*,\boldsymbol{y}^*),\, k\in\mathcal{K}}$ is optimal for USCTSNDP($\bar{\Phi}_\mathcal{K}$). Therefore, $\bar{\Phi}_\mathcal{K}$ is a finite feasible split satisfying the claimed optimal-cost equivalence, which proves Corollary 1. □

EC.2.3. Proof of Proposition 2

Proof Consider an optimal solution of the SCTSNDP, its corresponding flow splits for each commodity, and associated flow amount, delivery path and dispatch times for each flow split. Shift all the dispatch times of all flow splits as early as possible while keeping all flow split consolidations. Consequently, each dispatch time is set to the latest availability time among the split flows in the corresponding consolidation. This operation preserves the feasibility and the total cost because

the consolidations are unchanged and holding cost is zero. Then, each dispatch time at a node depends on the available times of the flow splits that originate at this node and on the arrival times of the flow splits that travel from other nodes and pass through this node. Suppose, for contradiction, there exists a dispatch time that is not of the form specified in the proposition. Let t be the earliest such dispatch time. Since t is not of the form specified in the proposition, it cannot be the availability time e^k of any commodity k and must be determined by the arrival time of some split flow. Therefore, there must exist a commodity k with a flow split $m \in M_k$, and an arc $a \in \mathcal{A}$ such that this flow split departs on arc a at time $t' = t - \tau_a$ and arrives at the current node at time t. Because of the choice of t and the positivity of travel times, t' must be one of the time points defined in the statement. Therefore, $t = t' + \tau_a$ is also one of the time points of the form defined in the statement, which contradicts its definition. This proves Proposition 2. □

EC.2.4. Proof of Theorem 1

Proof Let $(\boldsymbol{x}, \boldsymbol{y})$ be an optimal solution of formulation SSND$(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K})$ (i.e., an optimal SCTSNDP$(\mathcal{K})$ solution) of objective value z, where $\hat{\mathcal{T}}$ represents the complete discretization under $\hat{\Delta}$. For each commodity $k \in \mathcal{K}$, this solution induces a collection of flow splits $M_k(\boldsymbol{x}, \boldsymbol{y})$. Each split $m \in M_k(\boldsymbol{x}, \boldsymbol{y})$ is associated with a flow volume $g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ and a delivery timed path $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, such that $\sum_{m \in M_k(\boldsymbol{x},\boldsymbol{y})} g^{k,m}(\boldsymbol{x}, \boldsymbol{y}) = q^k$. Let $\bar{\mathcal{A}} = \{a = ((i,t),(j,t+\tau_{ij})) \in \mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}} : y^{t,t+\tau_{ij}}_{ij} > 0\}$ be the set of service arcs traversed by at least one flow split.

Given a partially time-expanded network $\underline{\mathcal{D}_{\mathcal{T}}}$ such that USSND$(\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K})$ is a relaxation of USCTSNDP$(\mathcal{K})$, it follows from Boland et al. (2017), Marshall et al. (2021), Shu et al. (2025) that, for every service arc $a \in \mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$, there exists a unique mapping arc $\tilde{a} = \pi(a) \in \underline{\mathcal{A}_{\mathcal{T}}}$, where a and $\tilde{a}$ are timed copies of the same physical arc (i,j), satisfying $\underline{a} = \underline{\tilde{a}} = (i,j)$. Here, $\underline{a}$ (resp. $\underline{\tilde{a}}$) denotes the physical arc in $\mathcal{A}$ underlying the timed arc a (resp. $\tilde{a}$). Further, any timed path P in $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ can be mapped to a unique feasible timed path P' in $\underline{\mathcal{D}_{\mathcal{T}}}$ by replacing each service arc of P with its corresponding mapping arc and adding appropriate holding arcs as needed. The arrival time of P' at the final node does not exceed that of P, and accordingly is a feasible timed path from the origin of commodity k to its destination.

Next, we show that, under the same partially time-expanded network $\underline{\mathcal{D}_{\mathcal{T}}}$, one can construct a feasible, though not necessarily optimal, solution $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}})$ to SSND$(\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K})$ with the same cost as the optimal solution $(\boldsymbol{x}, \boldsymbol{y})$ to SSND$(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K})$.

Using the same mapping rule $\pi : \mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}} \to \underline{\mathcal{A}_{\mathcal{T}}}$, for each commodity $k \in \mathcal{K}$ and each flow split $m \in M_k(\boldsymbol{x}, \boldsymbol{y})$, the path $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ is associated with a unique mapped path $\pi(P^{k,m}(\boldsymbol{x}, \boldsymbol{y}))$ in

$\underline{\mathcal{D}_\mathcal{T}}$ whose service arcs are $\pi(a)$ for all $a \in P^{k,m}(\boldsymbol{x},\boldsymbol{y})$ and is feasible for split m of commodity k. Aggregating these mapped timed paths defines the commodity-flow variables $\tilde{\boldsymbol{x}}$ as: $\widetilde{x}^k_{\tilde{a}} = \sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}:\pi(a)=\tilde{a}} x^k_a, \quad \forall \tilde{a} \in \underline{\mathcal{A}_\mathcal{T}}, k \in \mathcal{K}$. Similarly, the capacity variable $\widetilde{y}_{\tilde{a}}$ on each timed arc $\tilde{a} \in \underline{\mathcal{A}_\mathcal{T}}$ can be defined by summing the capacities in the optimal solution $(\boldsymbol{x},\boldsymbol{y})$ over all arcs that are mapped to $\tilde{a}$: $\widetilde{y}_{\tilde{a}} = \sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}:\pi(a)=\tilde{a}} y_a$.

Because each mapped timed path $\pi(P^{k,m}(\boldsymbol{x},\boldsymbol{y}))$ is a feasible timed path from the origin of commodity k to its destination, the flow conservation constraints and time-window requirements in the relaxation are satisfied. It remains to verify the capacity constraints. For any service arc $\tilde{a} \in \underline{\mathcal{A}_\mathcal{T}}$ associated with physical arc (i,j), we have $\sum_{k\in\mathcal{K}} q^k \widetilde{x}^k_{\tilde{a}} = \sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}:\pi(a)=\tilde{a}} \sum_{k\in\mathcal{K}} q^k x^k_a \le \sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}:\pi(a)=\tilde{a}} u_{ij} y_a = u_{ij}\widetilde{y}_{\tilde{a}}$, where the inequality follows from the capacity feasibility of $(\boldsymbol{x},\boldsymbol{y})$ in the complete formulation. Thus, $(\widetilde{\boldsymbol{x}},\widetilde{\boldsymbol{y}})$ is feasible for SSND($\underline{\mathcal{D}_\mathcal{T}},\mathcal{K}$).

Finally, the constructed solution $(\widetilde{\boldsymbol{x}},\widetilde{\boldsymbol{y}})$ has the same objective value as $(\boldsymbol{x},\boldsymbol{y})$. Indeed,

$$\begin{aligned}\sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}} f_{\underline{a}} y_a + \sum_{k\in\mathcal{K}}\sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}} (c^k_{\underline{a}} q^k) x^k_a &= \sum_{\tilde{a}\in\underline{\mathcal{A}_\mathcal{T}}}\sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}:\pi(a)=\tilde{a}} f_{\underline{\tilde{a}}} y_a + \sum_{k\in\mathcal{K}}\sum_{\tilde{a}\in\underline{\mathcal{A}_\mathcal{T}}}\sum_{a\in\mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}:\pi(a)=\tilde{a}} (c^k_{\underline{\tilde{a}}} q^k) x^k_a \\ &= \sum_{\tilde{a}\in\underline{\mathcal{A}_\mathcal{T}}} f_{\underline{\tilde{a}}}\widetilde{y}_{\tilde{a}} + \sum_{k\in\mathcal{K}}\sum_{\tilde{a}\in\underline{\mathcal{A}_\mathcal{T}}} (c^k_{\underline{\tilde{a}}} q^k)\widetilde{x}^k_{\tilde{a}}.\end{aligned}$$

The first equality follows from the uniqueness of the mapping $\pi(a)$ for each $a \in \mathcal{A}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ and the fact that each arc a and its mapped arc $\pi(a)$ correspond to the same physical arc. The second equality follows from the definitions of $\widetilde{y}_{\tilde{a}}$ and $\widetilde{x}^k_{\tilde{a}}$. Theorem 1 is therefore proved. □

EC.2.5. Proof of Lemma 1

Proof Consider a timed-cycle-free feasible solution $(\boldsymbol{x},\boldsymbol{y})$ to SSND($\underline{\mathcal{D}_\mathcal{T}},\mathcal{K}$), which yield the flow splits $M_k(\boldsymbol{x},\boldsymbol{y})$, the corresponding delivery timed path $P^{k,m}(\boldsymbol{x},\boldsymbol{y})$ and flow amount $g^{k,m}(\boldsymbol{x},\boldsymbol{y})$ for all $m \in M_k(\boldsymbol{x},\boldsymbol{y}), k \in \mathcal{K}$. As shown in definitions in Section 2.3, the set $\Gamma_k(\boldsymbol{x},\boldsymbol{y}) = \{\gamma^k_m\}_{m\in M_k(\boldsymbol{x},\boldsymbol{y})}$ where each γ^k_m is a copy of k and with $q^{\gamma^k_m} = g^{k,m}(\boldsymbol{x},\boldsymbol{y})$, constitutes a feasible demand disaggregation for commodity $k \in \mathcal{K}$. Aggregating these sets across all commodities yields the feasible split $\Phi_\mathcal{K}(\boldsymbol{x},\boldsymbol{y}) = \bigcup_{k\in\mathcal{K}} \Gamma_k(\boldsymbol{x},\boldsymbol{y})$ of $\mathcal{K}$.

We now show that $\mathcal{S}(\boldsymbol{x},\boldsymbol{y}) = (\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$, defined by (6)–(7), is feasible for USSND($\underline{\mathcal{D}_\mathcal{T}}, \Phi_\mathcal{K}(\boldsymbol{x},\boldsymbol{y})$) and has the same objective value as $(\boldsymbol{x},\boldsymbol{y})$. By construction, $S(\boldsymbol{x},\boldsymbol{y}) = (\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$ preserves the service arcs with positive flow in $(\boldsymbol{x},\boldsymbol{y})$, and each sub-commodity γ^k_m is routed along the same timed delivery path as flow split m of commodity k in $(\boldsymbol{x},\boldsymbol{y})$. Therefore, the flow-balance and time-window constraints are satisfied by $\bar{\boldsymbol{x}}$.

Moreover, since $\bar{\boldsymbol{y}} = \boldsymbol{y}$, the vehicle capacity installed on each timed arc remains unchanged, and so does the total fixed cost. For each timed arc $a \in \underline{\mathcal{A}_{\mathcal{T}}}$ on each path $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, we have $q^{\gamma_m^k} \bar{x}_a^{\gamma_m^k} = g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, which implies that $\sum_{k \in \mathcal{K}} q^k x_a^k = \sum_{k' \in \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})} q^{k'} \bar{x}_a^{k'}$ for all $a \in \underline{\mathcal{A}_{\mathcal{T}}}$. Hence, the capacity constraints are directly inherited from the feasibility of $(\boldsymbol{x}, \boldsymbol{y})$, and the total variable cost is also preserved. Thus, $S(\boldsymbol{x}, \boldsymbol{y})$ is feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) and has the same objective value as $(\boldsymbol{x}, \boldsymbol{y})$.

Suppose that $(\boldsymbol{x}, \boldsymbol{y})$ is optimal for SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$). Since feasible split of commodity set does not alter the feasible aggregate flows or the objective value of the continuous-flow formulation, SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) has the same optimal objective value as SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$). By the definition of the binary-flow formulation, USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) is obtained from SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) by restricting the flow variables to be binary. Hence, its optimal objective value cannot be further smaller. Since $S(\boldsymbol{x}, \boldsymbol{y})$ is feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$) and has the same cost as $(\boldsymbol{x}, \boldsymbol{y})$, $S(\boldsymbol{x}, \boldsymbol{y})$ is optimal for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y})$). □

EC.2.6. Proof of Theorem 2

proof Consider a timed-cycle-free feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ to SSND($\underline{\mathcal{D}_{\mathcal{T}}}, \mathcal{K}$) and its mapped solution $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ defined by (6)–(7). Let $M_k(\boldsymbol{x}, \boldsymbol{y})$ be the set of flow splits induced by $(\boldsymbol{x}, \boldsymbol{y})$, and let $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ and $g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ denote the timed path and the flow amount of split $m \in M_k(\boldsymbol{x}, \boldsymbol{y})$, respectively. By construction, $\Gamma_k(\boldsymbol{x}, \boldsymbol{y}) = \{\gamma_m^k\}_{m \in M_k(\boldsymbol{x},\boldsymbol{y})}$ with $q^{\gamma_m^k} = g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$, is a feasible demand disaggregation of k, and $\Phi_{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{y}) = \bigcup_{k \in \mathcal{K}} \Gamma_k(\boldsymbol{x}, \boldsymbol{y})$ is a feasible split of $\mathcal{K}$.

We first prove statement (i). By (6)–(7), each flow split $m \in M_k(\boldsymbol{x}, \boldsymbol{y})$ is mapped to a sub-commodity $\gamma_m^k \in \Gamma_k(\boldsymbol{x}, \boldsymbol{y})$ that has demand $q^{\gamma_m^k} = g^{k,m}(\boldsymbol{x}, \boldsymbol{y})$ and follows the same timed path $P^{k,m}(\boldsymbol{x}, \boldsymbol{y})$. Consequently, for every timed arc a and commodity $k \in \mathcal{K}$, $q^k x_a^k = \sum_{m \in M_k(\boldsymbol{x},\boldsymbol{y})} g^{k,m}(\boldsymbol{x}, \boldsymbol{y}) \bar{x}_a^{\gamma_m^k}$. Thus, $(\boldsymbol{x}, \boldsymbol{y})$ and $S(\boldsymbol{x}, \boldsymbol{y}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ use the same timed arcs, carry the same aggregate flow on each arc, and satisfy $\bar{\boldsymbol{y}} = \boldsymbol{y}$.

We now establish the equivalence in timed feasibility. Consider any discretization $\mathcal{T}' \subseteq \hat{\mathcal{T}}$. Because both solutions are associated with the same timed paths $\{P^{k,m}(\boldsymbol{x}, \boldsymbol{y})\}$, these paths are representable on $\underline{\mathcal{D}_{\mathcal{T}'}}$ for one solution if and only if they are representable for the other. Whenever they are representable in $\underline{\mathcal{D}_{\mathcal{T}'}}$, the incidence vector $\bar{\boldsymbol{x}}^{\gamma_m^k}$ of each path satisfies the flow-balance constraints for sub-commodity γ_m^k. Multiplying these constraints by $q^{\gamma_m^k}/q^k$ and summing over $\Gamma_k(\boldsymbol{x}, \boldsymbol{y})$ yields the flow-balance constraints for the original commodity k. Hence, the corresponding candidate solutions satisfy their respective flow-balance constraints simultaneously. Therefore,

$(\boldsymbol{x},\boldsymbol{y})$ satisfies the flow-balance constraints of SSND if and only if $S(\boldsymbol{x},\boldsymbol{y})$ satisfies the corresponding flow-balance constraints of the mapped USSND. Moreover, for every timed service arc a, $\sum_{k\in\mathcal{K}} q^k x_a^k = \sum_{k\in\mathcal{K}}\sum_{m\in M_k(\boldsymbol{x},\boldsymbol{y})} q^{\gamma_m^k}\bar{x}_a^{\gamma_m^k}$, and $\bar{y}_a = y_a$. Thus, $(\boldsymbol{x},\boldsymbol{y})$ satisfies the capacity constraint for arc a in SSND if and only if $S(\boldsymbol{x},\boldsymbol{y})$ satisfies the corresponding capacity constraint in the mapped USSND. It follows that $(\boldsymbol{x},\boldsymbol{y})$ is timed feasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}},\mathcal{K}$) if and only if $S(\boldsymbol{x},\boldsymbol{y})$ is timed feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}'}},\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$).

We then establish the equivalence in flat feasibility. Because each sub-commodity $\gamma_m^k \in \Gamma_k(\boldsymbol{x},\boldsymbol{y})$ corresponds to a split $m \in M_k(\boldsymbol{x},\boldsymbol{y})$, with quantity $q^{\gamma_m^k} = g^{k,m}$ and following the same timed path as the corresponding flow split, the two solutions have identical flat paths and arc-level consolidations. Specifically, the corresponding routing and consolidation plans of the USCTSNDP solution $S(\boldsymbol{x},\boldsymbol{y})$ can be represented respectively, by $\mathcal{P}(S(\boldsymbol{x},\boldsymbol{y})) = \{p^{k,m}(\boldsymbol{x},\boldsymbol{y})\}_{k\in\mathcal{K},\gamma_m^k\in\Gamma_k(\boldsymbol{x},\boldsymbol{y})}$ and $C(S(\boldsymbol{x},\boldsymbol{y})) = \{((i,j),\mathcal{J}_a(\boldsymbol{x},\boldsymbol{y},M_k(\boldsymbol{x},\boldsymbol{y}))) : a = ((i,t),(j,\bar{t})) \in \mathcal{A}^{\Delta}_{\mathcal{T}}\}$ with $\mathcal{J}_a(S(\boldsymbol{x},\boldsymbol{y})) = \{(k,m) : k\in\mathcal{K},\ \gamma_m^k\in\Gamma_k(\boldsymbol{x},\boldsymbol{y}),\ a\in P^{k,m}(\boldsymbol{x},\boldsymbol{y})\}$. Because $\gamma_m^k \in \Gamma_k(\boldsymbol{x},\boldsymbol{y})$ and $m \in M_k(\boldsymbol{x},\boldsymbol{y})$ are in one-to-one correspondence, it follows that $\mathcal{P}\big(\mathcal{S}(\boldsymbol{x},\boldsymbol{y})\big) = \mathcal{P}(\boldsymbol{x},\boldsymbol{y},M_k(\boldsymbol{x},\boldsymbol{y}))$ and $C\big(\mathcal{S}(\boldsymbol{x},\boldsymbol{y})\big) = C(\boldsymbol{x},\boldsymbol{y},M_k(\boldsymbol{x},\boldsymbol{y}))$. Thus, the two solutions have the same flat solution, which is representable on $\underline{\mathcal{D}_{\mathcal{T}'}}$ for one solution if and only if it is representable on this network for the other.

We next prove statement (ii). Suppose that $(\boldsymbol{x},\boldsymbol{y})$ is optimal for SSND($\underline{\mathcal{D}_{\mathcal{T}}},\mathcal{K}$). By Lemma 1, $S(\boldsymbol{x},\boldsymbol{y})$ is optimal for USSND($\underline{\mathcal{D}_{\mathcal{T}}},\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) and has the same cost as $(\boldsymbol{x},\boldsymbol{y})$. By Definition 3(iii), $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$) if and only if it is flat feasible under the complete discretization $\hat{\mathcal{T}}$. By statement (i), the condition holds if and only if $S(\boldsymbol{x},\boldsymbol{y})$ is flat feasible under $\hat{\mathcal{T}}$. Because $S(\boldsymbol{x},\boldsymbol{y})$ is optimal for the corresponding relaxation, Definition 3(iii) further implies that this condition holds if and only if $S(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$). Therefore, $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$) if and only if $S(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$). □

EC.2.7. Proof of Proposition 3

Proof For any commodity $k\in\mathcal{K}$, let $T^{k,m} = \sum_{a\in p^{k,m}(\boldsymbol{x},\boldsymbol{y})}\tau_a$ denote the total travel time of the flat path associated with flow split $m\in M_k(\boldsymbol{x},\boldsymbol{y})$. By the flow decomposition described in Section 4.3, the flow represented by the extracted origin-to-destination timed paths on each timed service arc does not exceed the corresponding arc flow in $\boldsymbol{x}^k$. Therefore, $\sum_{m\in M_k(\boldsymbol{x},\boldsymbol{y})}\frac{g^{k,m}(\boldsymbol{x},\boldsymbol{y})}{q^k}T^{k,m} \le \sum_{((i,t),(j,\bar{t}))\in\underline{\mathcal{A}_{\mathcal{T}}}}\tau_{ij}x_{ij}^{kt\bar{t}} \le l^k - e^k$, where the last inequality follows from Inequality (8). Moreover, since $\sum_{m\in M_k(\boldsymbol{x},\boldsymbol{y})}\frac{g^{k,m}(\boldsymbol{x},\boldsymbol{y})}{q^k} = 1$, there exists at least one $m\in M_k(\boldsymbol{x},\boldsymbol{y})$ such that $T^{k,m}\le l^k - e^k$. Hence, $p^{k,m}(\boldsymbol{x},\boldsymbol{y})$ is a k-feasible flat path. □

EC.2.8. Proof of Proposition 4

Proof By Proposition 3, for each commodity $k \in \mathcal{K}$, at least one flow split follows a k-feasible flat path. The repair procedure keeps the flow amount $g^{k,m}(\boldsymbol{x},\boldsymbol{y})$ of sub-commodity γ_m^k and replaces only paths that are not k-feasible with k-feasible ones. Therefore, every sub-commodity γ_m^k in $\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$ is routed from o^k to d^k within the corresponding time window, so the flow-balance and time-window constraints of USSND are satisfied. Moreover, since every sub-commodity associated with an original commodity k follows a k-feasible flat path, the repaired solution also satisfies (10). For any service arc $((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}$, (12) guarantees $\hat{y}_{ij}^{t\bar{t}}$ is a non-negative integer and satisfies the capacity constraints, therefore $\bar{S}(\boldsymbol{x},\boldsymbol{y}) = (\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ is feasible for USSND($\underline{\mathcal{D}_{\mathcal{T}}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$).

Furthermore, if $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$), then by Theorem 2(ii), $S(\boldsymbol{x},\boldsymbol{y})$ is flat feasible under the complete discretization. Hence, every sub-commodity γ_m^k in $S(\boldsymbol{x},\boldsymbol{y})$ must follow a k-feasible flat path. Therefore, the repair procedure does not perform path replacement. Equations (11) and (6) then give $\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}}$. Since $(\boldsymbol{x},\boldsymbol{y})$ is the optimal relaxation solution and all fixed costs are positive, no service arc carries more vehicles than needed for its load; otherwise, reducing the corresponding vehicle variable would keep feasibility and strictly lower the objective value. Hence $\bar{y}_{ij}^{t\bar{t}} = \lceil \sum_{k,m} q^{\gamma_m^k} \bar{x}_{ij}^{\gamma_m^k t\bar{t}} / u_{ij} \rceil$, which is exactly the value assigned by (12). Therefore, $\hat{\boldsymbol{y}} = \bar{\boldsymbol{y}}$, and consequently, $\bar{S}(\boldsymbol{x},\boldsymbol{y}) = S(\boldsymbol{x},\boldsymbol{y})$ □

EC.2.9. Proof of Theorem 3

Proof Consider an iteration of the SDDD algorithm, and let $(\boldsymbol{x},\boldsymbol{y})$ be an optimal solution to SSND($\underline{\mathcal{D}_{\mathcal{T}}},\mathcal{K}$), with objective value z. By Theorem 1, SSND($\underline{\mathcal{D}_{\mathcal{T}}},\mathcal{K}$) is a relaxation of SCTSNDP($\mathcal{K}$), and hence z is a valid lower bound. Since the optimality tolerance is set to zero, the stopping condition is satisfied only when $UB = LB$. Because LB and UB are valid lower and upper bounds, respectively, $UB = LB$ certifies that the incumbent upper-bound solution is optimal for SCTSNDP($\mathcal{K}$).

Suppose that $(\boldsymbol{x},\boldsymbol{y})$ is flat optimal for SCTSNDP($\mathcal{K}$), then there exists a feasible solution to SCTSNDP($\mathcal{K}$) with the same delivery paths, consolidation plan, and objective value z. By Lemma 1 and Theorem 2, the mapped solution $S(\boldsymbol{x},\boldsymbol{y})$ has the same cost as $(\boldsymbol{x},\boldsymbol{y})$ and is flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$). In this case, the repaired solution $\bar{S}(\boldsymbol{x},\boldsymbol{y})$ derived from (11)–(12) is identical to $S(\boldsymbol{x},\boldsymbol{y})$ by Proposition 4. Applying established upper-bound construction methods from the USCTSNDP literature to $\bar{S}(\boldsymbol{x},\boldsymbol{y})$ is therefore guaranteed to produce a feasible solution to

USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) with objective value z. This solution also corresponds to a feasible solution to SCTSNDP($\mathcal{K}$) with the same objective value, thereby certifying optimality. Consequently, the SDDD algorithm terminates with $LB = UB = z$.

Furthermore, if $(\boldsymbol{x},\boldsymbol{y})$ is not flat optimal for SCTSNDP($\mathcal{K}$), then, by Theorem 2, its mapped solution $S(\boldsymbol{x},\boldsymbol{y})$ is not flat optimal for USCTSNDP($\Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$). As described in Section 4.5, applying existing refinement procedures from the USCTSNDP literature to $S(\boldsymbol{x},\boldsymbol{y})$ is guaranteed to identify at least one node i at which a new time point $t \in \hat{\mathcal{T}}_i \setminus \mathcal{T}_i$, triggered by $S(\boldsymbol{x},\boldsymbol{y})$, is added to $\mathcal{T}_i$. The resulting discretization $\mathcal{T}'$ renders $S(\boldsymbol{x},\boldsymbol{y})$ either timed infeasible for USSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \Phi_{\mathcal{K}}(\boldsymbol{x},\boldsymbol{y})$) or flat infeasible on $\underline{\mathcal{D}_{\mathcal{T}'}}$. By the feasibility equivalence established in Theorem 2(i), this refinement likewise renders either $(\boldsymbol{x},\boldsymbol{y})$ timed infeasible for SSND($\underline{\mathcal{D}_{\mathcal{T}'}}, \mathcal{K}$) or flat infeasible on $\underline{\mathcal{D}_{\mathcal{T}'}}$. Because the complete discretization contains finitely many time points and each nonterminating iteration adds at least one new time point, the sequence of discretizations converges to the complete discretization within finitely many iterations, unless the algorithm terminates earlier. When the complete discretization is reached, SSND($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K}$) is the complete continuous-flow time-indexed model for SCTSNDP($\mathcal{K}$), as described in Section 2.2. Hence, its optimal solution is flat optimal for SCTSNDP($\mathcal{K}$), and the preceding argument yields $UB = LB$. Thus, the SDDD algorithm terminates after a finite number of iterations. □

EC.3. SDDD vs USDDD

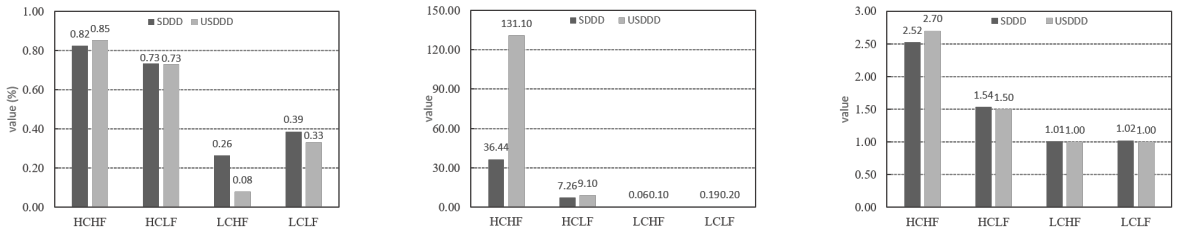
We compare SDDD (Algorithm 1) for the SCTSNDP($\mathcal{K}$) with the EDDD algorithm of Shu et al. (2025) applied to the corresponding USCTSNDP($\mathcal{K}$), denoted by USDDD. Both methods are evaluated on the 558 benchmark instances under the solver's default thread setting, when solving the corresponding MIP/IP models.

Figure EC.2 summarizes the average gap, runtime, and number of iterations for the two methods across the four instance groups. The results show that both methods solve all baseline instances within the prescribed optimality tolerance, with average runtimes and numbers of iterations of similar magnitude across the four groups. This suggests that the proposed SDDD framework handles split deliveries without substantially increasing computational difficulty relative to the corresponding unsplittable problem on these instances.

EC.4. Extension of SDDD to CTSNDP with additional operational cost

The computational experiments in Section 5.3 demonstrate the cost savings from splittable deliveries under the standard cost structure of the CTSNDP, which includes fixed costs incurred by

Figure EC.2 Comparison of SDDD for the SCTSNDP and USDDD for the corresponding USCTSNDP.



(a) Gap (b) Time (c) Iteration

operating vehicle dispatches and variable flow costs determined by shipment quantities and unit flow costs. However, in practice, these benefits may be reduced when additional operational costs are considered. For example, splittable deliveries may assign different portions of a commodity to later services or different terminals, thereby incurring holding and hub-transfer costs, which are additive in the transported flow. Established DDD algorithms are available for CTSNDP variants incorporating these two types of additional operational costs (Shu et al. 2024, He et al. 2024). For each of these two CTSNDP variants with additional operational cost, let VDDD denote the corresponding DDD algorithm developed by Shu et al. (2024) or He et al. (2024), respectively. Based on the corresponding VDDD, the SDDD framework can be adapted to evaluate the benefits of splittable deliveries under the corresponding operational-cost structure.

The corresponding SDDD starts from the initial time discretization and the partially time-expanded network defined by VDDD. The relaxation for the SCTSNDP is obtained by replacing the binary flow variables in the time-indexed relaxation model of VDDD with continuous variables. In each iteration, the relaxation is solved to obtain an optimal relaxation solution and update the lower bound of algorithm based on the optimal objective value. The obtained optimal relaxation solution then induces a feasible split of the original commodity set. Based on this feasible split, a mapped USCTSNDP instance, its relaxation and its mapped relaxation solution having the same cost as the SCTSNDP relaxation solution can be constructed. Based on the mapped relaxation solution, the upper-bound procedure of VDDD can then be applied to derive a valid upper bound for SDDD. When the gap between the updated upper bound and current lower bound reaches optimality tolerance, SDDD converges. Otherwise, the time-discretization refinement procedure of VDDD is applied to the mapped USCTSNDP instance based on the mapped relaxation solution, and SDDD proceeds to the next iteration.

The feasibility and optimality equivalences underlying SDDD remain valid for these two extensions. The additional costs affect only the objective function and therefore do not alter the feasibility equivalence. Moreover, because the holding and hub-transfer costs considered here are additive

in the transported flow, the mapping remains cost preserving: each flow split m and its corresponding sub-commodity γ_m^k have the same quantity and follow the same timed path, while the vehicle-dispatch variables remain unchanged. Hence, the mapped solutions incur identical fixed, flow, holding, and hub-transfer costs, and the optimality correspondence in Lemma 1 continues to hold. The flat-optimality equivalence established in Theorem 2 also remains valid because Lemma 1 continues to apply, and the additional costs do not affect flat feasibility.

EC.5. Interpretation and validity of the Inequalities (8) and (9)

To strengthen the relaxation model for the SCTSNDP, two valid inequalities are considered in Section 4.2. We will introduce their motivation, clarify their relationship to the corresponding USCTSNDP inequalities, and prove their validity for the SCTSNDP relaxation model as follows.

Inequalities (8) were originally proposed by Boland et al. (2017) for USCTSNDP to ensure that each commodity follows a *k-feasible* delivery path. We next show that these inequalities are still valid for SCTSNDP. Consider an optimal solution $(\boldsymbol{x}, \boldsymbol{y})$ to SSND$(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}, \mathcal{K})$. For each commodity $k \in \mathcal{K}$ and its corresponding flow splits $M_k(\boldsymbol{x}, \boldsymbol{y})$, let $T^{k,m}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{a \in p^{k,m}(\boldsymbol{x},\boldsymbol{y})} \tau_a$ denote the total travel time of its flat path. Because every flow split in an optimal SCTSNDP solution follows a k-feasible path, $T^{k,m}(\boldsymbol{x}, \boldsymbol{y}) \le l^k - e^k, \forall m \in M_k(\boldsymbol{x}, \boldsymbol{y})$ and $\sum_{m \in M_k(\boldsymbol{x},\boldsymbol{y})} g^{k,m}(\boldsymbol{x}, \boldsymbol{y}) = q^k$. Let $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$ denote the corresponding solution defined on the partially time-expanded network $\underline{\mathcal{D}_{\mathcal{T}}}$, obtained through the mapping used in the proof of Theorem 1. Since this mapping preserves the arcs traversed by every flow split, we have $\sum_{((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}} \tau_{ij} \underline{x}_{ij}^{kt\bar{t}} = \sum_{m \in M_k(\boldsymbol{x},\boldsymbol{y})} \frac{g^{k,m}(\boldsymbol{x},\boldsymbol{y})}{q^k} T^{k,m}(\boldsymbol{x}, \boldsymbol{y}), \forall k \in \mathcal{K}$. Therefore, $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$ satisfies Inequality (8), but only bound the flow-weighted average travel time across all utilized paths for commodity k.

For the USCTSNDP, Marshall et al. (2021) introduced $\left\lceil \frac{q^k}{u_{ij}} \right\rceil x_{ij}^{kt\bar{t}} \le y_{ij}^{t\bar{t}}, \forall\ ((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}, k \in \mathcal{K}$ to derive a lower bound on the number of vehicles required to serve a specific commodity. However, these inequalities are no longer valid when the flow variables are continuous. For example, if $q^k = 1.1$, $u_{ij} = 1$ and $x_{ij}^{kt\bar{t}} = 0.6$, the original inequalities will require at least two vehicles. However, the transported quantity can be accommodated by only one vehicle in practice. Therefore, we replace them with (9), which ensures that if commodity $k \in \mathcal{K}$ utilizes a service arc $((i,t),(j,\bar{t}))$, the number of vehicles assigned to that arc is at least the ratio of the commodity's flow to the arc capacity. We next establish their validity. Consider any commodity $k \in \mathcal{K}$ and any timed service arc $((i,t),(j,\bar{t})) \in \underline{\mathcal{A}_{\mathcal{T}}}$. If $q^k \le u_{ij}$, then the inequality reduced to $x_{ij}^{kt\bar{t}} \le y_{ij}^{t\bar{t}}$, which holds trivially if $x_{ij}^{kt\bar{t}} = 0$, otherwise, the capacity constraint and the integrality of $y_{ij}^{t\bar{t}}$ imply $y_{ij}^{t\bar{t}} \ge 1 \ge x_{ij}^{kt\bar{t}}$. If $q^k > u_{ij}$, the capacity constraint gives $q^k x_{ij}^{kt\bar{t}} \le u_{ij} y_{ij}^{t\bar{t}}$, and hence $\frac{q^k}{u_{ij}} x_{ij}^{kt\bar{t}} \le y_{ij}^{t\bar{t}}$.