The Continuous Time Service Network Design Problem

by

Amin Hosseininasab

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2015

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The service network design problem (SNDP) addresses the planning of operations for freight transportation carriers. Given a set of requests to transport commodities from specific origins to specific destinations, SNDP determines a continuous movement of vehicles to service demand. Demand becomes available for pick up at its origin by a given availability time and has to be dropped off at its destination by a given delivery deadline. The transportation plan considers matters of vehicle routing, consolidation, service schedule, empty vehicle repositioning, assignment of freight to operating vehicles, and vehicle stops and waiting times. The literature studies a periodic time approach to SNDP. This thesis generalizes the periodic time approach to SNDP by introducing a continuous time network and model. Several network and model reduction techniques are introduced, and a multi-cut Benders decomposition is developed to solve the continuous time model. To improve convergence of Benders decomposition, we strengthen the algorithm with a family of valid inequalities for SNDP. Numerical results show the benefits of the continuous time approach. Substantial reductions in computational effort and improved lower bounds are achieved by the multi-cut Benders decomposition algorithm.

Acknowledgements

Foremost, I would like to thank my supervisor, professor Fatma Gzara, who accepted me into University of Waterloo, and believed in me since day one. I am extremely grateful for her continuous and ever-present support throughout my Master's study, and for her patience, motivation, encouragement, understanding, and immense knowledge. I truly enjoyed and learned from my work with her, and owe most my knowledge in research to her guidance. I consider myself very fortunate to have worked under her supervision, and will forever be indebted to her.

My sincere thanks goes to Professor James Bookbinder, for his great help, advice, and for providing numerous Teaching Assistant opportunities for me to work with him. I truly appreciate all his kind efforts.

I would additionally like to thank Professor Bookbinder, and also Dr. Sibel Alumur for accepting to act as the readers of this thesis and providing useful feedbacks and comments.

My gratitude goes to my dearest friends in Waterloo. Ali Sarhadi, Ehsan Asadi, Forough Pourhossein, Majid Karimi, Meisam Soltani, and Alireza Rezvani, my Master's journey would not have been the same without our wonderful times together, and the enjoyable moments we've shared.

Last, but not least, my warmest gratitude goes to my family, and in particular to my parents. I would like to thank my farther for being the pillar and inspiration in my life. I have always looked up to you, and you have always been there when I needed a helping hand. Your words of advice and encouragement have inspired me on since as long as I can remember. I would not be anywhere near where I am today without your love, support, sacrifice, guidance, and the opportunities you provided for me throughout my life. I would like to thank my mother, for her love, sacrifices, prayers, and her countless dedications in my nurture and education. You have always been there in every stage of my life, and have warmly provided anything and everything I need. There are no words that can express my gratitude towards you both. Thank you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The service network design problem (SNDP) addresses the planning of operations for freight transportation carriers. The problem studies the design of the *service network*, where a transportation plan is determined for a planning horizon. A *service* is referred to the fulfilment of an order to transport a quantity of demand from a specified origin to a specified destination. A generalization of the pick up and delivery problem with time windows (PDPTW), the service network transportation plan not only determines vehicle routing and assignment of freight to operating vehicles but considers matters of consolidation, service schedule, empty vehicle repositioning, and vehicle stops and waiting times. Vehicles operate based on continuous movement in the network as opposed to moving to and from a depot in PDPTW. Continuous movement allows for the same vehicle to operate the same set of services in each planning horizon, hence the name *service network design*.

The SNDP falls into the context of a *network design* problem, where an optimized transportation plan is sought on a network of terminals. Freight becomes available at terminals during various times in the planning horizon and must be delivered to other terminals within a delivery deadline. This extends the tight time window assumption of terminals in PDPTW. The aim is to design a minimum cost network of operating vehicles which satisfy transportation demands. This is done by determining when and at what frequencies terminals are visited, the amount of time vehicles spend waiting at terminals in order to pick up a quantity of consolidated commodities, and the routes vehicles take on their course to pick up and deliver their allocated freight.

Service network design is an important and substantial problem faced by the freight transportation industry. Matters such as transportation cost, asset utilization and service quality are of vital significance. SNDP concerns any carrier consolidating freight for transport in origin-destination pairs; examples of which are less than truckload (LTL) carriers, express delivery services, and passenger transportation services. Such services may be operated by any of railroad, long distance maritime, ferry, air, or intermodal transportation modes. This has led to the increase of research in ways to model and solve the problem. However, SNDP has been proven to be very hard to solve. Solving relatively real-life size problems to optimality is not possible, and has provided the opportunity for research in determining efficient solution algorithms.

The literature poses SNDP as a periodic time problem. The planning horizon is divided into discrete time periods and it is assumed that all operations can occur in the defined periods only. Current models define a time-space network that repeats the physical structure of the terminal network for each time period. As a result, a node of the time-space network represents a specific terminal at a specific time period. The network thus incorporates the time dimension of the problem.

The periodic approach to SNDP comes with a number of disadvantages. A periodic time assumption must either use a very large number of periods to realistically model a real-life problem, or aggregate services into fewer time periods. A large number of periods increases the size of the resulting model and increases the difficulty of an already hard to solve problem. Aggregation into fewer number of periods reduces solution quality both in terms of cost and service. From a modeling perspective, the periodic SNDP restricts the ability to model stochastic transportation times. Any fluctuation in time must be a multiplier of the period duration, and so the number of periods must be high for a realistic analysis. The limitations of the periodic approach to SNDP provide the motivation for this thesis.

Our purpose is to develop a continuous time modeling approach for SNDP. To the best of our knowledge this is the first attempt in modeling SNDP in continuous time. We define a network independent of the time dimension and use it to formulate the continuous time model. Several reduction techniques are developed and applied to the continuous network and model. We build an exact solution methodology based on Benders decomposition. By exploiting the characterisitcs of SNDP we develop a family of valid inequalities, and use them to strengthen the Benders decomposition algorithm. Furthermore, we employ a multi-cut Benders decomposition approach. We test and compare the continuous model to the periodic model and show the advantages that may be gained from a continuous time perspective. We then test the proposed Benders decomposition approach and analyze the effect of developed improvements.

The thesis is organized as follows. Chapter 2 reviews the literature on the periodic time SNDP. In chapter 3 we develop the continuous time network and model. Benders decomposition is applied to the continuous time model in Chapter 4, and the relaxation of different sets of constraints is investigated. Chapter 5 presents algorithm improvements, and valid family of cuts for SNDP. Numerical results are given in Chapter 6, and the thesis is finally concluded in Chapter 7.

Chapter 2

Literature review

Early research on SNDP goes back to the work of Crainic and Rousseau (1986) and Farvolden and Powell (1994). Since then the literature has expanded in two main directions. The first group focus on generalizing SNDP by introducing more real-life aspects and complicating constraints. Sung and Song (2003) introduce cross docks into SNDP. Their approach does not explicitly consider a time setting and rather limits the maximum time a route can take. Meng and Wang (2011) study a hub and spoke structure with multiple vehicle types for SNDP. Andersen et al. (2009) study SNDP with coordination of multiple fleets. Fleets are defined as regionally different vehicles which also differ in type (e.g. train, ferry, etc.). Their intermodal model is deterministic, and tests are run on real life instances with 17 nodes and 84 time periods. They solve the model by a commercial solver and reach a gap of 10% after 24 hours with an improvement of 0.1% in the last 20 hours. The rest of the literature focuses on developing algorithmic methods to deal with the difficult and large size aspect of SNDP.

Two modeling approaches dominate the literature for SNDP. The first approach is referred to as the arc based formulation. The arc based formulation is modeled by commodity flow on arcs. The second approach is the path based formulation, where decision variables are defined on cycles. The path based formulation is shown to give higher LP bounds (Andersen et al., 2009). A different modeling approach is studied by Armacost et al. (2002). The authors introduce a composite variable formulation, and combine several routes into one composite route. Regardless of the modeling approach, SNDP is proven to be highly NP-hard (Ghamlouche et al., 2003).

Most research in solving SNDP focus on heuristic and meta-heuristic approaches. Examples are the works of Lai and Lo (2004) in ferry SNDP, Wang and Lo (2008) in multi-fleet passenger transportation, Teypaz et al. (2010) in large-scale SNDP, and Crainic et al. (2011) in progressive hedging-based meta-heuristics for SNDP. Kim et al. (1999) study the path based model for SNDP and develop several network reduction techniques. They study exact and approximation models and solve by a combination of column generation and heuristic procedures. They reduce the number of commodities by combining commodities with same origins to single super commodities. Their instances include 30 to 140 nodes, 4 to 9 hubs, 900 to 18,000 commodities aggregated into 30 to 130 product groups, and 5 to 7 vehicles. Pedersen et al. (2009) study the path based model of SNDP and solve by a neighbourhood tabu search algorithm. Their test instances include 20 to 100 nodes and 40 to 200 commodities.

Andersen et al. (2011) are the only authors that proposes an exact solution algorithm for SNDP. They develop a branch and price algorithm and iteratively generate cycles through two subproblems. In one subproblem they generate service cycles and in another they generate flow paths. Negative reduced cost cycles and paths are added to the linear master problem. They introduce acceleration techniques to form fast integer solutions. Tests are run on three approaches within a time limit of 10 hours. Test instances include 5 to 10 nodes with 10 to 50 services (possible number of paths from one terminal to another), 20 to 50 time periods, and 20 to 1000 commodities, with the largest instance having 10 nodes, 30 periods, 50 services and 800 commodities. Gaps of 0.9% to 21.5% are achieved but no instance is solved to optimality.

Lium et al. (2009) introduce uncertainty into SNDP. Demand is assumed uncertain and captured by the scenario based approach. The authors introduce an arc based model that uses outsourcing to satisfy unserviced demand. They show by small numerical examples of 6 nodes, 12 commodities and 7 time periods that stochasticity plays an important role in determining the optimal service routes. Hoff et al. (2010) study a stochastic SNDP with uncertain demand. They apply a path generation neighbourhood search heuristic to the SNDP formulation of Lium et al. (2009). Tests are run on small instances with 6 nodes and larger instances with 16 nodes, both with 14 commodities, 7 time periods, and 20 scenarios. Solutions were compared to an MIP solver which reached the optimal solution after 10 hours in the small instance and reached high gaps for large instances even after a week of computation.

Crainic et al. (2011) apply a Lagrangian relaxation to the stochastic arc based formulation of a restricted SNDP. They do not consider a time dimension and assume commodities may be picked up or delivered at any time. The master problem gives a set of open arcs which is used to determine the flow of commodities in the subproblems. The authors develop several Hedging based Meta-heuristic methods to produce service routes (open arcs). They test their approach on instances with 16 to 30 nodes, 14 to 80 commodities and 10 to 90 scenarios.

Bai et al. (2014) study the stochastic arc based formulation of SND. Stochasticity is defined on demand and is captured by scenarios. They expand the work of Lium et al. (2009) by incorporating the possibility to reroute previously determined services when demand is realized in each scenario. Tests analyze the effects of rerouting on highly uncertain instances with 6 nodes, 8 commodities, 5 periods and 20 scenarios. Larger instances with 20 commodities are tested but did not reach optimality. Results show that considering rerouting may lead to better overall solutions with lower costs and less need of outsourcing.

Table 2.1 presents an overview and classification of SNDP literature. The interested reader is referred to Crainic (2000) and Wieberneit (2008) for thorough reviews on SNDP.

Table 2.1: Classification of SNDP literature

Research	Environment	Time setting	Modeling approach	Solution approach
Crainic and Rousseau (1986)	Deterministic	No time dimension	path based	Heuristic
Farvolden and Powell (1994)	Deterministic	Periodic	Arc based	Subgradient-Heuristic
Kim et al. (1999)	Deterministic	Periodic	path based	Column generation-Heuristic
Armacost et al. (2002)	Deterministic	Periodic	path based	MIP solver
Sung and Song (2003)	Deterministic	Time limit	Arc based	Heuristic
Ghamlouche et al. (2003)	Deterministic	Periodic	path based	Heuristic
Lai and Lo (2004)	Deterministic	Periodic	Arc based	Heuristic
Wang and Lo (2008)	Deterministic	Periodic	Arc based	Heuristic
Lium et al. (2009)	Stochastic	Periodic	Arc based	MIP solver
Andersen et al. (2009)	Deterministic	Periodic	Arc based	MIP solver
Pedersen et al. (2009)	Deterministic	Periodic	path based	Heuristic
Hoff et al. (2010)	Stochastic	Periodic	Arc based	Heuristic
Teypaz et al. (2010)	Deterministic	Periodic	path based	Heuristic
Andersen et al. (2011)	Deterministic	Periodic	path based	Branch & price
Meng and Wang (2011)	Deterministic	Periodic	path based	MIP solver
Crainic et al. (2011)	Stochastic	No time dimension	Arc based	Lagrangian-Heuristic
Bai et al. (2014)	Stochastic	Periodic	Arc based	MIP solver

2.1 The periodic service network design problem

The periodic SNDP model is defined on a time-space network. This approach incorporates the time dimension of the problem into the network itself. The planning horizon is divided into $t = 1, 2, ..., |T|$ discrete time periods. The network is built by associating terminals to nodes and transportation routes to arcs; and repeating all nodes and arcs for each period. Figure 2.1 illustrates a time-space network with 3 terminals and 4 periods. Define a time-space network $G^p = (N^p, A^p)$, where N^p and A^p, denote the set of nodes and arcs, respectively. Superscript p denotes a periodic time network setting. A node $i \in N^p$ represents the physical location of terminal i at a specific period in the planning horizon. An arc $(i, j) \in A^p$ connects node i to j and represents a movement in time if i and j are associated with the same terminal($i = j$), and a movement in time and space otherwise. A cost c_{ij} is associated with arc (i, j), and represents the vehicle waiting cost if i and j are associated with the same terminal, or represents transportation cost otherwise.

Let K denote the set of commodities. A commodity $k \in K$ is defined as a request to move a quantity of d_k freight from origin $O(k)$ to destination $D(k)$. Freight becomes available at its origin at time $\sigma(k) \geq 0$ and must reach its destination by at most time $\tau(k) \leq T$. A set of homogeneous vehicles V, each with capacity η, is available to transport demand. Vehicles move in cycles; that is period T of the current planning horizon precedes period 1 of the next planning horizon. In other words, $t = 0$ is equivalent to $t = T$. The decision variables are defined as the integer service frequency variable x^t_{ij}, denoting the number of vehicles moving on arc (i, j) in period t; and continuous flow variables y^t_{ijk}, denoting the flow of commodity k on arc (i, j) in period t. The periodic SNDP model is formulated as below (Bia et al., 2014):

[PM1]

$$\min \quad \sum_{t\in T}\sum_{(i,j)\in A^p} c_{ij}x_{ij}^t \tag{2.1}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A^p} x_{ij}^t - \sum_{j:(j,i)\in A^p} x_{ji}^{t-1} = 0 \qquad \forall i \in N^p, \forall t \in T, \tag{2.2}$$

$$\sum_{k\in K} y_{ijk}^t \leq \eta x_{ij}^t \qquad \forall (i,j) \in A^p, i \neq j, \forall t \in T, \tag{2.3}$$

$$\sum_{j:(O(k),j)\in A^p} y_{O(k)jk}^{\sigma(k)} = d_k \qquad \forall k \in K, \tag{2.4}$$

$$\sum_{j:(j,D(k))\in A^p} y_{jD(k)k}^{\tau(k)-1} = d_k \qquad \forall k \in K, \tag{2.5}$$

$$\sum_{j:(i,j)\in A^p} y_{ijk}^t - \sum_{j:(j,i)\in A^p} y_{jik}^{t-1} = 0 \qquad \forall i \in N^p \setminus \{O(k), D(k)\}, \forall k \in K, \forall t \in T, \tag{2.6}$$

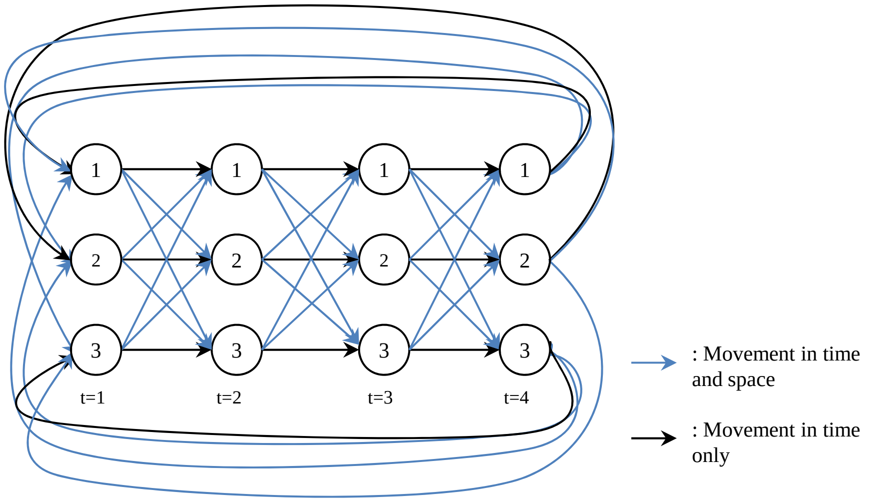
$$y_{ijk}^{\tau(k)} = 0 \qquad \forall (i,j) \in A^p, \forall k \in K \tag{2.7}$$

$$x_{ij}^t \geq 0, \quad \text{Integer} \qquad \forall (i,j) \in A^p, \forall t \in T, \tag{2.8}$$

$$y_{ijk}^t \geq 0 \qquad \forall (i,j) \in A^p, \forall k \in K, \forall t \in T. \tag{2.9}$$

Objective function (2.1) minimizes the sum of transportation and vehicle waiting costs. Constraints (2.2) ensure the balance of flow on the number of vehicles. Constraints (2.3) allow flow between two terminals only if there is vehicle movement, and also enforce an aggregated vehicle capacity. There is no capacity, or commodity flow restriction on arcs (i, j) if i and j are associated with the same terminal, which represent movement in time only. This setting allows a soft window on the pick up and delivery of commodities. A commodity may be picked up from its origin at any time after it becomes available, and may be delivered to its destination any time before its deadline. Constraints (2.4)-(2.6) enforce conservation of commodity flow, and constraints (2.7) ensure that no commodity flow takes place after its delivery deadline. Without constraints (2.7) the model may take advantage of the cyclic vehicle movements by picking up a commodity in the current planning horizon, and delivering it in the next planning horizon. We show in section 3.2.1 how these types of constraints are removed. Constraints (2.8), (2.9) are integer and non-negativity requirements on the decision variables.

Figure 2.1: The time-space network.



Commodity flow variables y^v_{ijk} are continuous. However, [PM1] results in integer values of y^v_{ijk} for vehicles $v \in V$. This is assured by setting vehicle capacity η to an integer multiplier of commodity volume. Currently, all commodities are assumed to consume the same physical volume, which then determines vehicle capacity. Vehicle capacity is set to the number of commodities that can fit into a vehicle $v \in V$. If commodities $k \in K$ have different physical capacities, variables y^v_{ijk} must be set to integer, and constraints (2.3) must be disaggregated by vehicle, for integer solutions of y^v_{ijk}.

An assumption of the current SNDP is that a commodity $k \in K$ may, with zero cost, change vehicle at any node along its path to destination. As vehicle flow variables x^t_{ij}

are integer, there is no differentiation between which vehicle is carrying which commodity. For example assume two vehicles entering node $i \in N^p$: $\sum\limits_{j \in A^p} x^t_{ji} = 2$ at time $t \in T$. Assume further that one of the vehicles is carrying load d_1, and the other is carrying load d_2. By the current formulation it is possible that the exiting vehicles from node i carry the loads of $d_1 + d_2/2$ and $d_2/2$. This results in $\eta - d_2/2$ available capacity for the second vehicle to pick up a different commodity $k \in K$, one which would not have been possible with $\eta - d_2$ available capacity. This is an unrealistic outcome. Not only is exchanging commodities between vehicles difficult ,and requires time and resources, but it also requires special terminals and infrastructure. The time and cost of exchanging commodities depends on where in the truck they are loaded; and rearranging a loaded truck is often difficult. Therefore, allowing such an option in order to possibly save on transportation costs is unrealistic.

On the other hand the model assumes that all terminals $i \in N^p$ can act as hub locations. A vehicle $v \in V$ is able to drop its load at one node $i \in N^p$ for another vehicle $v' \in V, v' \neq v$ to pick up at another time with no additional cost. Again this drop off-pick up activity is an unrealistic possible outcome. Not all nodes may have the infrastructure to allow storage, and assuming that all network nodes $i \in N^p$ are hub locations with no inventory cost is a major assumption. We therefore enforce that the flow of commodity k, y^t_{ijk}, assigned to a vehicle v, stays on that vehicle from origin to destination. Since commodity flow variables y^t_{ijk} are continuous, total commodity demand $d_k, k \in K$, may be assigned to multiple vehicles. Finally, if consolidation nodes or hub and spoke networks are sought, then the problem must specifically consider such nodes where vehicles may load or unload the commodities they carry with associated costs and times. This case is beyond the scope of this thesis.

To address the stated issues we must disaggregate vehicle movements. This is done by differentiating between vehicles, and replacing integer variables x_{ij}^t by binary variables x_{ij}^{tv}, which denote whether vehicle v uses arc (i,j) in period t. Consequently, variable y_{ijk}^t is replaced by variable y_{ijk}^{tv} which denotes the flow of commodity k on vehicle v moving on arc (i,j) at period t. The modified vehicle specific periodic model is formulated as follows:

[PM2]

$$\min \quad \sum_{t \in T} \sum_{v \in V} \sum_{(i,j) \in A^p} c_{ij} x_{ij}^{tv} \tag{2.10}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in A^p} x_{ij}^{tv} - \sum_{j:(j,i) \in A^p} x_{ji}^{t-1v} = 0 \qquad \forall i \in N^p, \forall t \in T, \forall v \in V, \tag{2.11}$$

$$\sum_{(i,j) \in A^p} x_{ij}^{Tv} \leq 1 \qquad \forall v \in V, \tag{2.12}$$

$$\sum_{k \in K} y_{ijk}^{tv} \leq \eta x_{ij}^{tv} \qquad \forall (i,j) \in A^p, i \neq j, \forall t \in T, \forall v \in V, \tag{2.13}$$

$$\sum_{v \in V} \sum_{j:(O(k),j) \in A^p} y_{O(k)jk}^{\sigma(k)v} = d_k \qquad \forall k \in K, \tag{2.14}$$

$$\sum_{v \in V} \sum_{j:(D(k),j) \in A^p} y_{jD(k)k}^{\tau(k)-1v} = d_k \qquad \forall k \in K, \tag{2.15}$$

$$\sum_{j} y_{ijk}^{tv} - \sum_{j} y_{jik}^{t-1v} = 0 \qquad \forall i \in N^p \setminus \{O(k), D(k)\}, \forall t \in T, \forall k \in K, \forall v \in V, \tag{2.16}$$

$$\sum_{v \in V} \sum_{(i,j) \in A^p} y_{ijk}^{\tau(k)v} = 0 \qquad \forall k \in K, \tag{2.17}$$

$$x_{ij}^{tv} \geq 0, \quad \text{Binary} \qquad \forall (i,j) \in A^p, \forall t \in T, \forall v \in V, \tag{2.18}$$

$$y_{ijk}^{tv} \geq 0 \qquad \forall (i,j) \in A^p, \forall k \in K, \forall t \in T, \forall v \in V. \tag{2.19}$$

Objective function (2.10) and Constraints (2.11), are similar to Objective function (2.1) and Constraints (2.2) in [PM1], respectively. Constraints (2.12) ensure that a vehicle $v \in V$ is used at most once. This constraint is not required in the previous formulation as vehicle flow is aggregated over all vehicles $v \in V$. Constraints (2.13) account for disaggregated capacity constraints on vehicles $v \in V$. Constraints (2.14), (2.15) are similar to Constraints

(2.4), (2.5) in [PM1]. Constraints (2.16) enforce commodity flow balance and do not allow the exchange of commodities $k \in K$ between vehicles $v \in V$ or their drop off at any node $i \in N^p, i \neq D(k)$. An advantage of the vehicle specific model is the possibility of using vehicles with different capacities. The downside is that the problem increases in size by the number of vehicles $|V|$, and possibly makes it more difficult to solve.

This concludes the review of the periodic approach to SNDP. The next chapter takes a continuous time approach to SNDP, and constructs the continuous time network and model.

Chapter 3

The continuous service network design problem

The time-space network incorporates the time dimension of SNDP into the network itself. To the best of our knowledge the time-space network is the only approach available to SNDP. However, this approach makes a discrete time assumption, and occurrences in time must be assigned to specific time points. All schedules and vehicle movements must be based on the assumed period duration, and any time in between periods must be aggregated to the discrete time periods. This setting also removes the possibility of considering stochastic transportation durations, as any change to transportation time must be at least the duration of a whole time period.

The time-space network approach also comes with the price of repeating the original network by the number of periods considered. Assuming that the original network $G = (N, A)$ has $|N|$ nodes and $|A| = |N|(|N|-1)$ arcs, the time space network has $|N^p| = |N||T|$ nodes and $|A^p| = |N|^3|T|$ arcs. As the number of periods increases, the size of the time-space network increases. This increase limits the possibility of using short time periods over a planning horizon. As an example, if we were to design the service network with 1 hour periods on a horizon of 30 days, we must repeat the entire network $24 \times 30 = 720$ times. Such large networks exceed the capacities of available solution approaches. The current literature assumes long time periods, and aggregates the operations between periods into far apart points of time, which reduces the accuracy and quality of solution for real-life operations.

We propose a commodity-based network for SNDP. Our aim is to incorporate the time dimension of the problem into the model, rather than the network. We assume continuous time which is a generalization of the periodic time representation.

3.1 Constructing the continuous time network

To model the time dimension of SNDP we introduce time variables and determine the time a vehicle $v \in V$ arrives or departs terminals $i \in N$. Arrival (departure) variables can only take one value, meaning that a vehicle $v \in V$ may arrive (depart) a terminal $i \in N$ at most once. However, the problem may require a vehicle to visit a terminal more than once. Assuming that transportation costs and times have the triangular property, there is only one case where a vehicle $v \in V$ may be required to visit the a terminal $i \in N$ more than once. When a terminal $i \in N$ serves as the origin $O(k)$ or destination $D(k)$ to more than one commodity $k \in K$, a vehicle $v \in V$ may choose to visit i to pick up a commodity $k \in K$, and visit i again to deliver another commodity $k' \in K, k' \neq k$. To remove this case, we model the problem on the following network.

We define the continuous time network on a directed graph $G^c = (N^c, A^c)$ where N^c is the set of nodes and A^c is the set of arcs, respectively. Superscript c represents a continuous time network setting. The set of nodes N^c is built by associating a node $i \in N^c$ to the origin $O(k)$ of a commodity $k \in K$, and a node $i \in N^c$ to the destination $D(k)$ of a commodity $k \in K$, resulting in $2|K|$ nodes. In this setting a vehicle $v \in V$ visits a node $i \in N^c$ either to pick up or deliver a commodity $k \in K$. On the other hand, not only would a vehicle $v \in V$ visit a node $i \in N^c$ at most once, but if it does, it picks up or delivers a commodity $k \in K$. Visiting a node $i \in N^c$ without performing a service only increases cost and is not optimal. A node $i \in N^c$ is associated with a terminal location in the original network, and multiple nodes may share the same physical location. A node $i \in N^c$ is associated with waiting cost c_i, which represents the cost of waiting at the associated terminal of node i. Graph G^c is complete and all nodes $i \in N^c$ are connected to all other nodes $j \in N^c, j \neq i$, by arcs of set A^c. An arc $(i, j) \in A^c$ is associated with cost c_{ij}, and time t_{ij}, which represent transportation cost and time if nodes i and j are not associated with the same terminal, and are set to zero otherwise. The final step in forming the continuous time network is ensuring that vehicles start at some terminal, perform a number of pick up and deliveries, and return to the same starting terminal. Such vehicle cycles have the characteristics below:

1. A cycle starts at an origin node $O(k), k \in K$, as no delivery is possible without pick up.

2. A cycle ends at a destination node $D(K), k \in K$, as no pick up is possible without delivery in the same planning horizon.

Using these two facts we define "end of horizon" nodes $D'(k)$, which represent the associated terminals of all commodity destinations $D(k), k \in K$ at the end of the planning horizon. These nodes are added to N^c and denoted by $N^c_s \subset N^c$. Based on its associated terminal, a destination $D(k)$ is connected to its corresponding end of horizon node $D'(k)$, by adding arc $(D(k), D'(k))$ to A^c, with cost $c_{D(k)D'(k)} = 0$ and time $t_{D(k)D'(k)} = 0$. A node $i \in N_s^c$ has the same waiting cost c_i as its corresponding terminal, and is only used to indicate the location of a vehicle $v \in V$, at the end of the cycle. A node $i \in N_s^c$ is then connected to all origin nodes $O(k'), k' \in K$ by "end of horizon" arcs $(D'(k), O(k'))$, that are added to set A^c. An arc $(D'(k), O(k'))$ indicates that a vehicle $v \in V$ starts at origin $O(k')$ and ends at destination $D(k)$.

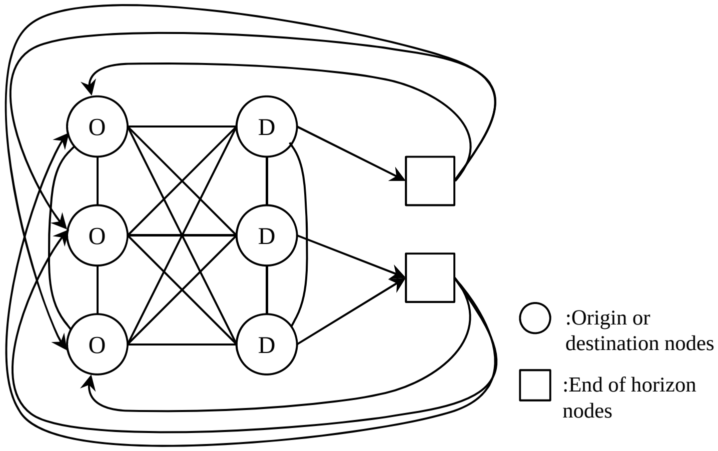
The cost $c_{D'(k)O(k')}$ and time $t_{D'(k)O(k')}$ associated with end of horizon arcs $(D'(k), O(k')$, is defined in two ways. If $D'(k)$ and $O(k')$ are associated with the same terminal, then arc $(D'(k), O(k')$ is associated a cost $c_{D'(k)O(k')} = 0$ and transportation time $t_{D'(k)O(k')} = -T$; otherwise the associated cost is $c_{D'(k)O(k')} = c_{D(k)O(k')}$ and transportation time is $t_{D'(k)O(k')} = -T + t_{D(k)O(k')}$. The transportation time of the end of horizon arcs is what distinguishes them from regular transportation arcs, and indicates moving to the start of the next planning horizon. Note that as the arrival time to any node is non-negative, the use of end of horizon arcs is only possible at a time greater than $-t_{D'(k)O(k')}$. Figure 3.1 displays a continuous time network for $|K| = 3$, such that the last two destinations are associated with one terminal.

By the above definitions, the number of nodes is at most $|N^c| = 3|K|$, and the set of arcs is at most $|A^c| = 5|K|^2 - |K|$ arcs. The continuous model may be modeled on G^c. However G^c is general and not all arcs $(i, j) \in A^c$ are needed. The network G^c can be considerably reduced by the characteristics of SNDP, as shown in the next section.

3.1.1 Network reduction

The network G^c is general and contains a large number of arcs. A node $i \in N^c$ may be connected to as many as $|N^c| - 1$ nodes, meaning that there could be $|N^c| - 1$ nodes a

Figure 3.1: The continuous time network.



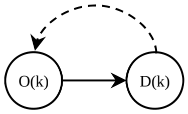
vehicle $v \in V$ may visit when leaving node i. We simplify the network by removing arcs of set A^c that cannot be in a feasible cycle. This is done by exploiting the relationship between commodities $k \in K$, and their respective origins $O(k)$ and destinations $D(k)$ in N^c.

The first set of redundant arcs are of form $(D(k), O(k)), k \in K$. The reasoning is simple: after we have serviced and delivered a commodity $k \in K$ we never have to visit $O(k)$. Therefore, direct movement from the destination of a commodity to its origin is infeasible. By this we remove all arcs $(D(k), O(k)), k \in K$ from the arc set A^c, as illustrated by Figure 3.2. Throughout this thesis, a dashed line indicates an eliminated arc. The second set of redundant arcs are shown by Lemma 1:

Lemma 1. *Given $\tau(k) \leq \sigma(k'), k, k' \in K$, a feasible cycle cannot contain the arcs $(O(k), D(k')), (O(k'), D(k)), (O(k), O(k')), (O(k'), O(k)), (D(k), D(k')), (D(k'), D(k)),$ $(D(k'), O(k))$.*

Proof. Recall that if origin $O(k), k \in K$ is visited, commodity k is picked up; and if

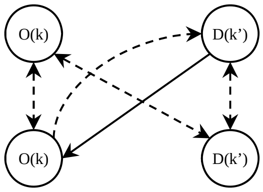
Figure 3.2: Removing arcs of form $(D(k), O(k))$.



destination $D(k), k \in K$ is visited, commodity k is delivered. As $\tau(k) \leq \sigma(k')$ then commodity k has to be serviced before commodity k' becomes available. That is, both $O(k)$ and $D(k)$ must be visited before $O(k')$ and $D(k')$ in any feasible path. Therefore, the only direct movement between the two commodities' nodes could be moving from $D(k)$ to $O(k')$. □

By Lemma 1 we remove all arcs $(O(k), D(k')), (O(k'), D(k)), (O(k), O(k')), (O(k'), O(k))$, $(D(k), D(k')), (D(k'), D(k)), (D(k'), O(k))$ from A^c, where $\tau(k) \leq \sigma(k'), k, k' \in K$. This is shown by Figure 3.3.

Figure 3.3: Removing arcs by Lemma 1.



We further reduce network G^c using the time relationship of nodes $i, j \in N^c$ such that i and j are associated with the same terminal. We show the reduction by the three cases below:

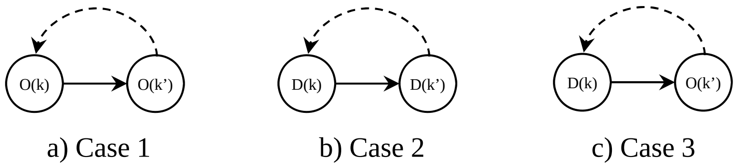
1. Consider two origins $O(k), O(k'), k, k' \in K$ such that $O(k), O(k')$ are associated with the same terminal and $\sigma(k) \leq \sigma(k')$. Since commodity k becomes available before

commodity k', it is safe to assume that in a consecutive visit of a vehicle $v \in V$ to origins $O(k), O(k')$, origin $O(k)$ is visited first, and $O(k')$ is visited second. By this assumption we remove arc $(O(k'), O(k))$ from A^c, as shown by Figure 3.4.a.

2. Consider two destinations $D(k), D(k'), k, k' \in K$ such that $D(k), D(k')$ are associated with the same terminal and $\tau(k) \leq \tau(k')$. Since commodity k must be delivered before commodity k', then it is safe to assume that in a consecutive visit of a vehicle $v \in V$ to destinations $D(k), D(k')$, destination $D(k)$ is visited first, and $D(k')$ is visited second. By this assumption we remove arc $(D(k'), D(k))$ from A^c, as shown by Figure 3.4.b.

3. Consider origins $O(k), k \in K$, and destination $D(k'), k' \in K, k' \neq k$, such that $O(k), D(k')$ are associated with the same terminal. It is safe to assume that in a consecutive visit to $O(k)$ and $D(k)$, the delivery of commodity k' takes place before the pick up of commodity k. That is in a consecutive visit of a vehicle $v \in V$ to origin $O(k)$ and destination $D(k')$, destination $D(k')$ is visited first, and $O(k)$ is visited second. By this assumption we remove arc $(O(k), D(k'))$ from A^c, as shown by Figure 3.4.c.

Figure 3.4: Removing arcs between nodes associated with the same terminal.



This concludes the construction of the continuous time network. Section 3.2 develops the continuous time model.

3.2 Developing the continuous time formulation of SNDP

We define the continuous time formulation on network $G^c = (N^c, A^c)$, constructed and reduced in Section 3.1. The continuous time model consists of two parts. The first part models commodity and vehicle flow constraints and the second parts models time constraints. Let binary decision variables x_{ij}^v denote vehicle movement and indicate whether vehicle $v \in V$ moves on arc (i,j). Let continuous decision variables y_{ijk}^v denote commodity flow and indicate the quantity of commodity $k \in K$ moving on arc (i,j) on vehicle $v \in V$. Transportation cost is calculated as $\sum\limits_{v \in V} \sum\limits_{(i,j) \in A^c} c_{ij} x_{ij}^v$. Commodity and vehicle flow constraints are as follows.

$$\sum_{j:(i,j) \in A^c} x_{ij}^v - \sum_{j:(j,i) \in A^c} x_{ji}^v = 0 \qquad \forall i \in N^c, \forall v \in V \quad (3.1)$$

$$\sum_{(i,j) \in A^c} x_{ij}^v \leq 1 \qquad \forall v \in V, \quad (3.2)$$

$$\sum_{k \in K} y_{ijk}^v \leq \eta x_{ij}^v \qquad \forall (i,j) \in A^c, \forall v \in V, \quad (3.3)$$

$$\sum_{v \in V} \sum_{j:(O(k),j) \in A^c} y_{O(k)jk}^v = d_k \qquad \forall k \in K, \quad (3.4)$$

$$\sum_{v \in V} \sum_{j:(j,D(k)) \in A^c} y_{jD(k)k}^v = d_k \qquad \forall k \in K, \quad (3.5)$$

$$\sum_{j:(i,j) \in A^c} y_{ijk}^v - \sum_{j:(j,i) \in A^c} y_{jik}^v = 0 \qquad \forall i \in N^c \setminus \{O(k), D(k)\}, \forall k \in K, \forall v \in V, \quad (3.6)$$

$$\sum_{v \in V} \sum_{(i,j) \in A^c : j \in N_s^c} y_{ijk}^v = 0 \qquad \forall k \in K, \quad (3.7)$$

$$x_{ij}^v \quad \text{binary} \qquad \forall (i,j) \in A^c, \forall v \in V, \quad (3.8)$$

$$y_{ijk}^v \geq 0 \qquad \forall (i,j) \in A^c, \forall k \in K, \forall v \in V. \quad (3.9)$$

Constraints (3.1) address the vehicle balance requirements. Constraints (3.2) are similar to constraints (2.12), and only allow one cycle per vehicle. Constraints (3.3) enforce vehicle capacity and flow of commodity only if there is vehicle movement on set A^c. Constraints (3.4)-(3.6) are conservation of commodity flow. Constriants (3.7) ensure that no commodity flow takes place on end of horizon arcs $(i, j) \in A^c, j \in N_s^c$, and constraints (3.9), and (3.15) are binary and non-negative requirements on the decision variables.

The time dimension may be modeled in two ways. The first approach introduces variables u_i^v and s_i^v as the arrival and departure times of vehicle $v \in V$, to and from node $i \in N^c$. Vehicle waiting cost is calculated as $\sum\limits_{i \in N^c} \sum\limits_{v \in V} c_i(s_i^v - u_i^v)$, and the time constraints are written as:

$$u_i^v \leq s_i^v \qquad \forall i \in N^c, \forall v \in V, \quad (3.10)$$

$$t_{ij}x_{ij}^v - (1 - x_{ij}^v)T \leq u_j^v - s_i^v \leq t_{ij}x_{ij}^v + (1 - x_{ij}^v)T \qquad \forall (i, j) \in A^c, \forall v \in V, \quad (3.11)$$

$$s_i^v \leq T \sum_{j:(i,j) \in A^c} x_{ij}^v \qquad \forall i \in N^c, \forall v \in V, \quad (3.12)$$

$$s_{O(k)}^v \geq \sigma(k) \sum_{j:(O(k),j) \in A^c} x_{O(k)j}^v \qquad \forall k \in K, \forall v \in V, \quad (3.13)$$

$$u_{D(k)}^v \leq \tau(k) \qquad \forall k \in K, \forall v \in V, \quad (3.14)$$

$$s_i^v, u_i^v \geq 0 \qquad \forall i \in N^c, \forall v \in V. \quad (3.15)$$

Constraints (3.10) ensure that vehicle arrival precedes vehicle departure. Transportation time is implemented by constraints (3.11). Constraint (3.11) accounts for the time it takes to move on arc (i, j) if it is used by vehicle v ($x_{ij}^v = 1$), and becomes redundant if there is no vehicle movement ($x_{ij}^v = 0$). Constraints (3.12) set the departure time of unvisited nodes to zero. Together with (3.10), constraints (3.12) set arrival time of unvisited nodes to zero. Constraints (3.13) and (3.14) ensure that the availability and delivery deadline of all commodities $k \in K$ are respected. Constraints (3.8) are non-negative requirements on the time decision variables.

The second approach to model time constriants is to define variables w^v_{ij} as the time vehicle $v \in V$ departs node $i \in N^c$, heading for node $j \in N^c : (i,j) \in A^c_1$. In this case vehicle waiting cost is calculated as $\sum\limits_{i \in N^c} \sum\limits_{v \in V} c_i \left(\sum\limits_{j \in N^c} w^v_{ij} - \sum\limits_{j \in N^c} \left(w^v_{ji} + t_{ji} x^v_{ji} \right) \right)$, and time constraints are written as:

$$\sum_{j:(j,i) \in A^c} \left(w^v_{ji} + t_{ji} x^v_{ji} \right) \leq \sum_{j:(i,j) \in A^c} w^v_{ij} \qquad \forall i \in N^c, \forall v \in V, \quad (3.16)$$

$$w^v_{O(k)i} \geq \sigma(k) x_{O(k)i} \qquad \forall i \in N^c : (O(k), i) \in A^c, \forall k \in K, \forall v \in V, \quad (3.17)$$

$$w^v_{iD(k)} + t_{iD(k)} x^v_{iD(k)} \leq \tau(k) x^v_{iD(k)}, \qquad \forall i \in N^c : (i, D(k)) \in A^c_1, \forall k \in K, \forall v \in V, \quad (3.18)$$

$$w^v_{ij} \leq T x^v_{ij} \qquad \forall (i,j) \in A^c, \forall v \in V, \quad (3.19)$$

$$w^v_{ij} \geq 0 \qquad \forall (i,j) \in A^c, \forall v \in V. \quad (3.20)$$

Constraints (3.16) set the arrival time of a vehicles $v \in V$ to a node $i \in N^c$ to precede its departure time. Moreover Constraints (3.16) keep track of time spent by a vehicle $v \in V$ in the network. In other words, these constraints serve the same purpose as constraints (3.10) and (3.11) together in the previous definition of time variables. Constraints (3.17) and (3.18) enforce the availability and delivery deadline of the commodities, and constraint (3.19) sets the departure time (and consequently the arrival time) of unvisited nodes to zero.

Both modelling approaches give the same solution to the continuous time SNDP. However the feasible region, when relaxing the binary requirements on variables x^v_{ij}, is different. Based on our initial experiments, the second approach of defining time variables w^v_{ij} outperforms the first approach by computational time and tightness of the relaxed feasible region. The advantage of the second approach is significantly clear in large size instances. Therefore we pursue and use only the second approach in this thesis. The complete model for the continuous time SNDP is:

[CM1]

$$\min \quad \sum_{(i,j)\in A^c}\sum_{v\in V} c_{ij}x_{ij}^v + \sum_{i\in N^c}\sum_{v\in V} c_i\left(\sum_{j\in N^c} w_{ij}^v - \sum_{j\in N^c}\left(w_{ji}^v + t_{ji}x_{ji}^v\right)\right) \tag{3.21}$$

$$\text{s.t.} \quad (3.1) - (3.9), (3.16) - (3.20).$$

Model [CM1] can be reduced for an easier solution. This reduction is discussed in the following section.

3.2.1 Model reduction

We reduce [CM1] by distinguishing the arcs of set A^c. We show how this distinction leads to a smaller and reduced model [CM2]; and prove by Theorem 1 how [CM1] and [CM2] are equivalent. We define subsets A_1^c, A_2^c, and A_3^c, such that all arcs $(i,j) \in A^c$ fall into one subset only, and $A^c = A_1^c \cup A_2^c \cup A_3^c$. Each set is defined as follows:

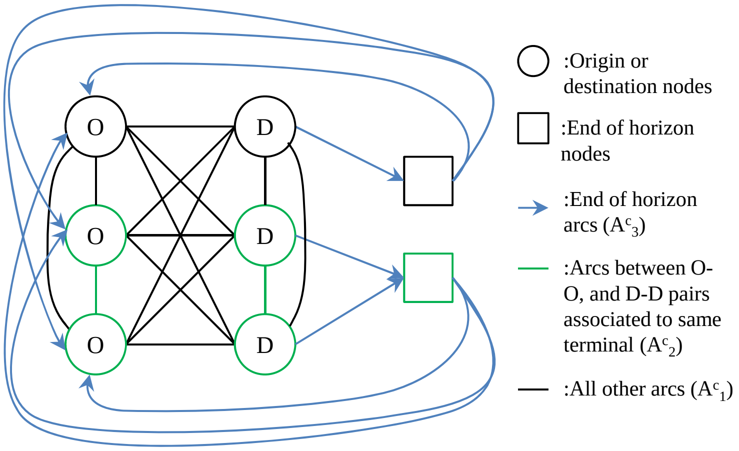
1. Let $A_1^c \subset A^c$ denote the set of arcs (i,j), such that i and j are not associated with the same terminal; or i and j are associated with the same terminal but take forms of $(O(k), D(k')), (D(k), O(k')), k, k' \in K$.

2. Let $A_2^c \subset A^c$ denote the set of arcs (i,j), such that i and j are associated with the same terminal, and are of the forms $(O(k), O(k')), (D(k), D(k')), k, k' \in K$.

3. Let $A_3^c \subset A^c$ denote the set of end of horizon arcs (i,j), where $i \vee j \in N_s^c$.

By this definition we only define vehicle flow for set $A_1^c \cup A_3^c$, and only define commodity flow for set $A_1^c \cup A_2^c$. This reduces both the number of vehicle flow variables x_{ij}^v, and time

Figure 3.5: Definition of arc sets for the continuous time network.


O
O
O
D
D
D
:Origin or destination nodes
:End of horizon nodes
:End of horizon arcs (A^c_3)
:Arcs between O-O, and D-D pairs associated to same terminal (A^c_2)
:All other arcs (A^c_1)

variables w_{ij}^v by $|V||A_2^c|$; and reduces the number of commodity flow variables y_{ijk}^v by $|V||A_3^c|$. In addition to the number of variables, the number of constraints in [CM2] are reduced compared to [CM1]. Model [CM2] is formulated as below.

[CM2]

$$\min \quad \sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V} c_{ij}x_{ij}^v + \sum_{i\in N^c}\sum_{v\in V} c_i\left(\sum_{j\in N^c} w_{ij}^v - \sum_{j\in N^c}\left(w_{ji}^v + t_{ji}x_{ji}^v\right)\right) \tag{3.22}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A_1^c\cup A_3^c} x_{ij}^v - \sum_{j:(j,i)\in A_1^c\cup A_3^c} x_{ji}^v = 0 \qquad \forall i\in N^c, \forall v\in V, \tag{3.23}$$

$$\sum_{(i,j)\in A_1^c\cup A_3^c} x_{ij}^v \le 1 \qquad \forall v\in V, \tag{3.24}$$

$$\sum_{k\in K} y_{ijk}^v \le \eta x_{ij}^v \qquad \forall (i,j)\in A_1^c, \forall v\in V, \tag{3.25}$$

$$\sum_{v\in V}\sum_{j:(O(k),j)\in A_1^c\cup A_2^c} y_{O(k)jk}^v = d_k \qquad \forall k\in K, \tag{3.26}$$

$$\sum_{v\in V}\sum_{j:(j,D(k))\in A_1^c\cup A_2^c} y_{jD(k)k}^v = d_k \qquad \forall k\in K, \tag{3.27}$$

$$\sum_{j:(i,j)\in A_1^c \vee (i,j)\in A_2^c, j=D(k)} y_{ijk}^v - \sum_{j:(j,i)\in A_1^c \vee (j,i)\in A_2^c, j=O(k)} y_{jik}^v = 0$$

$$\forall i \in N^c \setminus \{O(k), D(k)\}, \forall k \in K, \forall v \in V, \tag{3.28}$$

$$\sum_{j:(j,i)\in A_1^c \cup A_3^c} (w_{ji}^v + t_{ji} x_{ji}^v) \leq \sum_{j:(i,j)\in A_1^c \cup A_3^c} w_{ij}^v \qquad \forall i \in N^c, \forall v \in V, \tag{3.29}$$

$$w_{O(k)i}^v \geq \sigma(k) x_{O(k)i} \qquad \forall i \in N^c : (O(k), i) \in A_1^c \cup A_3^c, \forall k \in K, \forall v \in V, \tag{3.30}$$

$$w_{iD(k)}^v + t_{iD(k)} x_{iD(k)}^v \leq \tau(k) x_{iD(k)}^v$$

$$\forall i \in N^c : (i, D(k)) \in A_1^c \cup A_3^c, \forall k \in K, \forall v \in V, \tag{3.31}$$

$$w_{ij}^v \leq T x_{ij}^v \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V, \tag{3.32}$$

$$w_{ij}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V, \tag{3.33}$$

$$x_{ij}^v \quad \text{binary} \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V, \tag{3.34}$$

$$y_{ijk}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_2^c, \forall k \in K, \forall v \in V. \tag{3.35}$$

Objective function (3.22) corresponds to objective function (3.21) with no vehicle flow on arcs $(i,j) \in A_2^c$. Constraints (3.23), (3.24), and (3.29)-(3.32) correspond to Constraints, (3.1), (3.2), and (3.16)-(3.20), respectively, with no vehicle flow on arcs A_2^c. Constraints (3.3) only enforce vehicle capacity on arcs of set A_1^c. In other words commodities may flow without the requirement of vehicle flow on arcs A_2^c (there is no commodity flow on arcs A_3^c). Constraints (3.26), (3.27) are supply and demand constraints defined on arcs $A_1^c \cup A_2^c$. Constraints (3.28) are conservation of commodity flow. In addition to conserving commodity flow on arcs $(i,j) \in A_1^c$; for a commodity $k \in K$ these constraints conserve flow on arcs $(i, D(k)), (O(k), i) \in A_2^c$. By constraints (3.26)-(3.28) we ensure that a commodity $k' \in K, k' \neq k$ does not move on arcs $(i, D(k)), (O(k), i) \in A_2^c$, and that they may only be used for commodity k. If commodities k' were allowed to flow on these arcs, a vehicle may unrealistically drop a commodity at node $O(k) : (O(k), j) \in A_2^c$ (or node $i : (i, D(k)) \in A_2^c$), and pick it up later in the cycle at node $j : (O(k), j) \in A_2^c$ (node $D(k) : (i, D(k)) \in A_2^c$)

using arc $(i, D(k))$ (arc $(O(k), i)$) as a short cut. There is no commodity flow defined on subset A_3^v, and so constraints (3.7) are automatically satisfied. We now prove by Theorem 1 that solving [CM1] and [CM2] is equivalent.

Theorem 1. *Solving model [CM2] is equivalent to solving model [CM1].*

Proof. An arc $(i, j) \in A_2^c$ corresponds to $c_{ij} = 0$. Therefore, solutions $x_{ij}^v = 1, (i, j) \in A_2^c$ and $x_{ij}^v = 0, (i, j) \in A_2^c$ give the same value in objective (3.21) in [CM1]. The proof reduces to proving feasibility. We show that if a solution $\hat{x}_{ij}^v$ satisfies constraints (3.23)-(3.33), it satisfies constraints (3.1)-(3.9), (3.16)-(3.20). It is clear that if $\hat{x}_{ij}^v$ satisfies vehicle and commodity flow conservation enforced by Constraints (3.23)-(3.28), it satisfies conservation of flow in Constraints (3.1)-(3.9). Now, Constraints (3.29)-(3.32) enforce time constraints on set arc set $A_1^c \cup A_3^c$; and Constraints (3.16)-(3.20) enforce time constraints on arc set $A^c = A_1^c \cup A_2^c \cup A_3^c$. An arc $(i, j) \in A_2^c$ falls in one of the below cases:

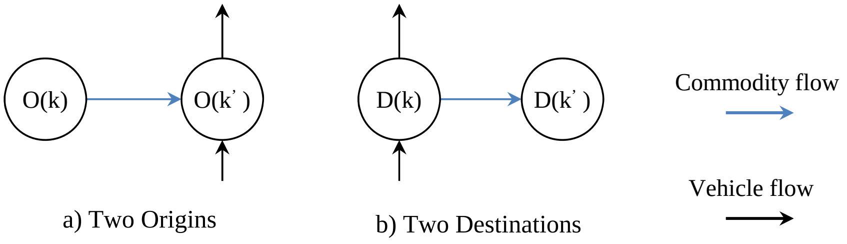
1. If $(i, j) \in A_2^c$, and $i = O(k), j = O(k'), k, k' \in K$, we have $\sigma(k) \leq \sigma(k')$ by network reduction techniques. If commodity k flows on arc $(O(k), O(k'))$, it is picked up at $O(k')$, without requiring vehicle movement on arc $(O(k), O(k'))$. If the vehicle picking up commodity k satisfies its availability time $\sigma(k)$ then it is feasible to Constraints (3.16)-(3.20). In this case we have $x_{O(k')i} = 1$ for some $i \in N^c$, therefore:
   $w_{O(k')i} \geq \sigma(k')$ by constraint (3.30).
   As $\sigma(k) \leq \sigma(k')$ we thus have:
   $w_{O(k')i} \geq \sigma(k)$.
   which means time constraints (3.16)-(3.20) are satisfied for commodity k without requiring vehicle movement to $O(k)$.

2. If $(i, j) \in A_2^c$, and $i = D(k), j = D(k'), k, k' \in K$, we have $\tau(k) \leq \tau(k')$ by network reduction techniques. If commodity k' flows on arc $(D(k), D(k'))$, it is dropped off

at $D(k)$, without requiring vehicle movement on arc $(D(k), D(k'))$. If the vehicle dropping off commodity k' satisfies its delivery deadline $\tau(k')$ then it is feasible to Constraints (3.16)-(3.20). In this case we have $x_{iD(k)} = 1$ for some $i \in N^c$, therefore: $w_{iD(k)} + t_{iD(k)} \leq \tau(k)$ by constraint (3.31).
As $\tau(k) \leq \tau(k')$ we thus have:
$w_{iD(k)} + t_{iD(k)} \leq \tau(k')$.
which means time constraints (3.16)-(3.20) are satisfied for commodity k' without requiring vehicle movement to $D(k')$.

Hence, commodity flow of arcs $(i, j) \in A_2^c$ is satisfied by Constraints (3.4)-(3.6). This concludes the proof. Figure 3.6 shows a pick up or delivery service without requiring vehicle movement. □

Figure 3.6: Simultaneous pick up or delivery of commodities sharing their associated terminal.



By the reduction in size, solving [CM2] is easier than [CM1]. However, [CM2] is still hard to solve using a commercial solver. We develop a decomposition algorithm using Benders decomposition to aid in solving [CM2]. This solution algorithm is presented in Chapter 4.

Chapter 4

Solution by Benders decomposition

Model [CM2] presented in Chapter 3 is difficult (or even impossible in large size problems) to solve using commercial solvers alone. An approach to solving large size problems is the use of decomposition schemes. Branch and price algorithms are employed in the literature to solve the periodic SNDP. However, as the linear relaxation lower bound is poor in SNDP models, the branch and price algorithm is not very efficient and requires a lot of improvement. We propose using Benders decomposition.

Benders decomposition generally solves a relaxed master problem [RMP] with a set of relaxed constraints in the first stage. The relaxed constraints are moved into recourse subproblems which are bound by the solution of [RMP]. Based on the solutions of the subproblems, feasibility and optimality cuts are derived and added to [RMP]. The procedure continues until all subproblems are feasible and give the same objective bound as the master problem. Given a set of vehicle movements $\hat{x}_{ij}^{v}$, [CM2] may be decomposed in three ways:

1. Commodity flow constraints (3.25)-(3.28) are relaxed and moved into the subproblem.

2. Time constraints (3.29)-(3.32) are relaxed and moved into the subproblem.
3. Both commodity flow and time constraints (3.25)-(3.32) are relaxed and moved into the subproblem.

4.1 Relaxing commodity flow constraints

Model [CM2] may be decomposed by moving commodity flow constraints (3.25)-(3.28) and (3.35) to [SPy], and reformulating [CM2] on a set of feasibility constraints Ω_y^{feas}. We have the reformulation of model [CM2] as [MPy]:

[MPy]

$$\min \quad \sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V} c_{ij}x_{ij}^v + \sum_{i\in N^c}\sum_{v\in V} c_i\left(\sum_{j\in N^c} w_{ij}^v - \sum_{j\in N^c}\left(w_{ji}^v + t_{ji}x_{ji}^v\right)\right) \tag{4.1}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A_1^c\cup A_3^c} x_{ij}^v - \sum_{j:(j,i)\in A_1^c\cup A_3^c} x_{ji}^v = 0 \qquad \forall i\in N^c, \forall v\in V, \tag{4.2}$$

$$\sum_{j:(i,j)\in A_1^c\cup A_3^c} x_{ij}^v \leq 1 \qquad \forall i\in N^c, \forall v\in V, \tag{4.3}$$

$$\sum_{(i,j)\in A_1^c\cup A_3^c : i\in N_s^c} x_{ij}^v \leq 1 \qquad \forall v\in V, \tag{4.4}$$

$$\sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V} \Theta_{ij}^{r^y} x_{ij}^v \geq \Theta_0^{r^y} \qquad \forall r^y\in\Omega_y^{\text{feas}}, \tag{4.5}$$

$$\sum_{j:(j,i)\in A_1^c\cup A_3^c} \left(w_{ji}^v + t_{ji}x_{ji}^v\right) \leq \sum_{j:(i,j)\in A_1^c\cup A_3^c} w_{ij}^v \qquad \forall i\in N^c, \forall v\in V, \tag{4.6}$$

$$w_{O(k)i}^v \geq \sigma(k)x_{O(k)i} \qquad \forall i\in N^c : (O(k),i)\in A_1^c\cup A_3^c, \forall k\in K, \forall v\in V, \tag{4.7}$$

$$w_{iD(k)}^v + t_{iD(k)}x_{iD(k)}^v \leq \tau(k)x_{iD(k)}^v \qquad \forall i\in N^c : (i,D(k))\in A_1^c\cup A_3^c, \forall k\in K, \forall v\in V, \tag{4.8}$$

$$w_{ij}^v \leq T x_{ij}^v \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V, \quad (4.9)$$

$$w_{ij}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V, \quad (4.10)$$

$$x_{ij}^v \quad \text{binary} \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V. \quad (4.11)$$

where Ω_y^{feas} is the set of feasibility cuts that ensure the solution of [MPy] is feasible for commodity transportation. The coefficients Θ_{ij}^{vry} are used to describe such feasibility cuts. The set Ω_y^{feas} is unknown beforehand. We start by solving a relaxed master problem [RMPy] defined on $\bar{\Omega}_y^{\text{feas}} \subset \Omega_y^{\text{feas}}$. The solution of the relaxed master problem [RMPy] $\hat{x}_{ij}^v$, gives a set of vehicle cycles that satisfy the availability time and delivery deadlines of all commodities $k \in K$. To check feasibility of $\hat{x}_{ij}^v$ for commodity flow constraint we solve subproblem [SPy].

[SPy]

$$\min \quad 0$$

$$\text{s.t.} \quad \sum_{k \in K} y_{ijk}^v \leq \eta \hat{x}_{ij}^v \qquad \forall (i,j) \in A_1^c, \forall v \in V, \quad (4.12)$$

$$\sum_{v \in V} \sum_{j:(O(k),j) \in A_1^c \cup A_2^c} y_{O(k)jk}^v = d_k \qquad \forall k \in K, \quad (4.13)$$

$$\sum_{v \in V} \sum_{j:(j,D(k)) \in A_1^c \cup A_2^c} y_{jD(k)k}^v = d_k \qquad \forall k \in K, \quad (4.14)$$

$$\sum_{j:(i,j) \in A_1^c \vee (i,j) \in A_2^c, j=D(k)} y_{ijk}^v - \sum_{j:(j,i) \in A_1^c \vee (j,i) \in A_2^c, j=O(k)} y_{jik}^v = 0$$

$$\forall i \in N^c \setminus \{O(k), D(k)\}, \forall k \in K, \forall v \in V, \quad (4.15)$$

$$y_{ijk}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_2^c, \forall k \in K, \forall v \in V. \quad (4.16)$$

Subproblem [SPy] is a capacitated multi-commodity flow problem, restricted by open arcs in $\hat{x}_{ij}^v$. It determines the flow of commodities given vehicle movements $\hat{x}_{ij}^v$ from [RMPy]. As [SPy] has no objective, we only seek feasibility in the subproblem, and the first feasible solution of [RMPy] in [SPy] is the optimal solution to [CM2]. If [SPy] is infeasible we derive a Benders feasibility cut and add it to the set $\bar{\Omega}_y^{\text{feas}}$. To derive the feasibility cuts we take the dual of [SPy]. Associating dual variables $\alpha_{ij}^v, \beta_{O(k)}, \beta_{D(k)}, \beta_{ik}^v$ with constriants (4.12)-(4.15) we have the dual of [SPy] as:

[DSPy]

$$\begin{aligned}
\max \quad & -\sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V}\eta\hat{x}_{ij}^v\alpha_{ij}^v + \sum_{k\in K} d_k(\beta_{O(k)}+\beta_{D(k)}) && && (4.17)\\
\text{s.t.} \quad & -\alpha_{O(k)j}^v + \beta_{O(k)} - \beta_{jk}^v \le 0 && \forall (O(k),j)\in A_1^c, j\neq D(k), \forall k\in K, \forall v\in V, && (4.18)\\
& -\alpha_{O(k)D(k)}^v + \beta_{O(k)} - \beta_{D(k)} \le 0 && \forall (O(k),D(k))\in A_1^c, \forall k\in K, \forall v\in V, && (4.19)\\
& \beta_{O(k)} - \beta_{jk}^v \le 0 && \forall (O(k),j)\in A_2^c, \forall k\in K, \forall v\in V, && (4.20)\\
& -\alpha_{D(k)j}^v + \beta_{D(k)} - \beta_{jk}^v \le 0 && \forall (D(k),j)\in A_1^c, \forall k\in K, \forall v\in V, && (4.21)\\
& \beta_{D(k)} - \beta_{jk}^v \le 0 && \forall (D(k),j)\in A_2^c, \forall k\in K, \forall v\in V, && (4.22)\\
& -\alpha_{ij}^v - \beta_{jk}^v \le 0 && \forall (i,j)\in A_1^c, i,j\notin\{O(k),D(k)\}, \forall k\in K, \forall v\in V, && (4.23)\\
& -\alpha_{iO(k)}^v + \beta_{ik}^v - \beta_{O(k)} \le 0 && \forall (i,O(k))\in A_1^c, \forall k\in K, \forall v\in V, && (4.24)\\
& -\alpha_{iD(k)}^v + \beta_{ik}^v - \beta_{D(k)} \le 0 && \forall (i,D(k))\in A_1^c, i\neq O(k), \forall k\in K, \forall v\in V, && (4.25)\\
& \beta_{ik}^v - \beta_{O(k)} \le 0 && \forall (i,O(k))\in A_2^c, \forall k\in K, \forall v\in V, && (4.26)\\
& \beta_{ik}^v - \beta_{D(k)} \le 0 && \forall (i,D(k))\in A_2^c, \forall k\in K, \forall v\in V, && (4.27)\\
& \alpha_{ij}^v \ge 0 && \forall (i,j)\in A_1^c, \forall v\in V. && (4.28)
\end{aligned}$$

The dual subproblem [DSPy] is always feasible as the all-zero solution is a feasible answer. Whenever [DSPy] is unbounded, [SPy] is infeasible. To remove the infeasible solution x_{ij}^v from [RMPy] we derive the unbounded dual ray $(\alpha_{ij}^{vr}, \beta_{O(k)}^r, \beta_{D(k)}^r)$ and add cut (4.29) to $\bar{\Omega}_y^{\text{feas}}$:

$$-\sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V}\eta\alpha_{ij}^{vr}x_{ij}^v + \sum_{k\in K} d_k(\beta_{O(k)}^r + \beta_{D(k)}^r) \le 0. \tag{4.29}$$

The procedure continues by adding feasibility cuts (4.29) to [RMPy] until [DSPy] finds a feasible solution to subproblem [SPy], which is the optimal solution to [CM2].

4.2 Relaxing time constraints

Model [CM2] may be decomposed by moving time constriants (3.29)-(3.32) and (3.15) to subproblem [SPvt] which is disaggregated by vehicles $v \in V$, and reformulating [CM2] on a set of feasibility cuts Ω_t^{feas} and optimality cuts Ω_t^{opt}. Associating value function variables z^v with a subproblem [SPvt], we have the reformulated master problem [MPt] as:

[MPt]

$$\min \quad \sum_{(i,j)\in A_1^c \cup A_3^c} \sum_{v\in V} c_{ij} x_{ij}^v + \sum_{v\in V} z^v \tag{4.30}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A_1^c \cup A_3^c} x_{ij}^v - \sum_{j:(j,i)\in A_1^c \cup A_3^c} x_{ji}^v = 0 \qquad \forall i \in N^c, \forall v \in V, \tag{4.31}$$

$$\sum_{(i,j)\in A_1^c \cup A_3^c : i \in N_s^c} x_{ij}^v \leq 1 \qquad \forall v \in V, \tag{4.32}$$

$$z^v + \sum_{(i,j)\in A_1^c \cup A_3^c} \Lambda_{ij}^{vh} x_{ij}^v \geq \Lambda_0^{vh} \qquad \forall v \in V, \forall h \in \Omega^{\text{opt}}, \tag{4.33}$$

$$\sum_{(i,j)\in A_1^c \cup A_3^c} \Theta_{ij}^{vr^t} x_{ij}^v \geq \Theta_0^{vr^t} \qquad \forall v \in V, \forall r^t \in \Omega_t^{\text{feas}}, \tag{4.34}$$

$$\sum_{k\in K} y_{ijk}^v \leq \eta x_{ij}^v \qquad \forall (i,j) \in A_1^c, \forall v \in V, \tag{4.35}$$

$$\sum_{v\in V} \sum_{j:(O(k),j)\in A_2^c} y_{O(k)jk}^v = d_k \qquad \forall k \in K, \tag{4.36}$$

$$\sum_{v\in V} \sum_{j:(j,D(k))\in A_2^c} y_{jD(k)k}^v = d_k \qquad \forall k \in K, \tag{4.37}$$

$$\sum_{j:(i,j)\in A_1^c \vee (i,j)\in A_2^c, j=D(k)} y_{ijk}^v - \sum_{j:(j,i)\in A_1^c \vee (j,i)\in A_2^c, j=O(k)} y_{jik}^v = 0$$
$$\forall i \in N^c \setminus \{O(k), D(k)\}, \forall k \in K, \forall v \in V, \tag{4.38}$$

$$x_{ij}^v \quad \text{binary} \qquad \forall (i,j) \in A_1^c \cup A_3^c, \forall v \in V, \tag{4.39}$$

$$y_{ijk}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_2^c, \forall k \in K, \forall v \in V. \tag{4.40}$$

where Ω_t^{opt} is the set of cuts that bound variables z^v depending on vehicle movements x_{ij}^v and Ω_t^{feas} is the set of cuts that ensure the solution of [RMPy] is feasible in terms of time constraints. The coefficients Λ_{ij}^{vh} and $\Theta_{ij}^{vr^t}$ are used to describe such cuts. Sets Ω_t^{feas} and Set Ω_t^{opt} are unknown beforehand. We start by solving a relaxed master problem [RMPy] defined on sets $\bar{\Omega}_t^{\text{feas}} \subset \Omega_t^{\text{feas}}$, and $\bar{\Omega}_t^{\text{opt}} \subset \Omega_t^{\text{opt}}$. The solution of the relaxed master problem [RMPt], $\hat{x}_{ij}^v$, gives a set of vehicle and commodity routes that satisfy demand and commodity flow requirements. These routes may however be infeasible in terms of when commodities become available and when they are needed at their destination. To determine the feasibility and optimality of $\hat{x}_{ij}^v$ to [CM2], we solve subproblem [SPvt]. Subproblem [SPvt] determines the vehicle arrival and departure times given vehicle movements $\hat{x}_{ij}^v$. We have the [SPvt] formulation as:

[SPvt]

$$\min \quad \sum_{i \in N^c} \sum_{v \in V} c_i \left(\sum_{j \in N^c} w_{ij}^v - \sum_{j \in N^c} \left(w_{ji}^v + t_{ji} \hat{x}_{ji}^v \right) \right) \tag{4.41}$$

$$\text{s.t.} \quad \sum_{j:(j,i) \in A_1^c \cup A_3^c} \left(w_{ji}^v + t_{ji} \hat{x}_{ji}^v \right) \leq \sum_{j:(i,j) \in A_1^c \cup A_3^c} w_{ij}^v \qquad \forall i \in N^c, \tag{4.42}$$

$$w_{O(k)i}^v \geq \sigma(k) \hat{x}_{O(k)i} \qquad \forall i \in N^c : (O(k), i) \in A_1^c \cup A_3^c, \forall k \in K, \tag{4.43}$$

$$w_{iD(k)}^v + t_{iD(k)} \hat{x}_{iD(k)}^v \leq \tau(k) \hat{x}_{iD(k)}^v \qquad \forall i \in N^c : (i, D(k)) \in A_1^c \cup A_3^c, \forall k \in K, \tag{4.44}$$

$$w_{ij}^v \leq T \hat{x}_{ij}^v \qquad \forall (i,j) \in A_1^c \cup A_3^c, \tag{4.45}$$

$$w_{ij}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_3^c, \tag{4.46}$$

If [SPvt] is infeasible we derive a Benders feasibility cut and add it to set $\bar{\Omega}_t^{\text{feas}}$. If [SPvt] is feasible but not optimal we derive a Benders optimality cut and add it to $\bar{\Omega}_t^{\text{opt}}$. To derive the optimality and feasibility cuts we take the dual of [SPvt]. Associating dual variables

$\pi_i^v, \gamma_{ik}^v, \theta_{ik}^v, \lambda_{ij}^v$ with constraints (4.42)-(4.45) we have the dual of [SPvt] as:

[DSPvt]

$$
\begin{aligned}
\max \quad & \sum_{i \in N^c} \left(\sum_{j:(j,i) \in A_1^c \cup A_3^c} t_{ji} \hat{x}_{ji}^v \right) \pi_i^v + \sum_{i:(O(k),i) \in A_1^c \cup A_3^c} \sum_{k \in K} \left(\sigma(k) \hat{x}_{O(k)i}^v \right) \gamma_{ik}^v \\
& + \sum_{i:(i,D(k)) \in A_1^c \cup A_3^c} \sum_{k \in K} \left(\left(-\tau(k) + t_{iD(k)} \right) \hat{x}_{iD(k)} \right) \theta_{ik}^v - T \sum_{(i,j) \in A_1^c \cup A_3^c} \hat{x}_{ij}^v \lambda_{ij}^v && (4.47) \\
& \pi_i^v - \pi_{O(k)}^v - \lambda_{iO(k)}^v \leq c_i - c_{O(k)} \qquad \forall (i, O(k)) \in A_1^c \cup A_3^c, \forall k \in K, && (4.48) \\
& \pi_{O(k)}^v - \pi_i^v + \gamma_{ik}^v - \lambda_{O(k)i}^v \leq c_{O(k)} - c_i \\
& \qquad \forall (O(k), i) \in A_1^c \cup A_3^c, i \neq D(k), \forall k \in K, && (4.49) \\
& \pi_i^v - \pi_{D(k)}^v - \theta_{ik}^v - \lambda_{iD(k)}^v \leq c_i - c_{D(k)} \\
& \qquad \forall (i, D(k)) \in A_1^c \cup A_3^c, i \neq O(k), \forall k \in K, && (4.50) \\
& \pi_{D(k)}^v - \pi_i^v - \lambda_{D(k)i}^v \leq c_{D(k)} - c_i \qquad \forall (D(k), i) \in A_1^c \cup A_3^c, \forall k \in K, && (4.51) \\
& \pi_{O(k)}^v - \pi_{D(k)}^v - \theta_{O(k)k}^v + \gamma_{D(k)k}^v - \lambda_{O(k)D(k)}^v \leq c_{O(k)} - c_{D(k)} \\
& \qquad \forall (O(k), D(k)) \in A_1^c \cup A_3^c, \forall k \in K, && (4.52) \\
& \pi_i^v - \pi_j^v - \lambda_{ij}^v \leq c_i - c_j \qquad \forall (i,j) \in A_1^c \cup A_3^c, i, j \notin \{O(k), D(k)\}, \forall k \in K, && (4.53) \\
& \pi_i^v \geq 0 \qquad \forall i \in N^c, && (4.54) \\
& \theta_{ik}^v \geq 0 \qquad \forall (i, D(k)) \in A_1^c \cup A_3^c, \forall k \in K, && (4.55) \\
& \gamma_{ik}^v \geq 0 \qquad \forall (O(k), i) \in A_1^c \cup A_3^c, \forall k \in K, && (4.56) \\
& \lambda_{ij}^v \geq 0 \qquad \forall (i,j) \in A_1^c \cup A_3^c. && (4.57)
\end{aligned}
$$

Subproblem [DSPvt] is always feasible since setting $\pi_i^v = c_i, \forall i \in N^c$ and all other variables to zeros is a feasible answer. If [DSPvt] is unbounded, then [SPvt] is infeasible. To remove the infeasible solution x_{ij}^v we derive the unbounded dual ray $(\pi_i^{vr}, \theta_{ik}^{vr}, \gamma_{ik}^{vr}, \lambda_{ij}^{vr})$, and add the infeasibility cuts (4.58) to Ω_t^{feas}:

$$\begin{aligned}&\sum_{i\in N^c}\pi_i^{vr}\sum_{j:(j,i)\in A_1^c\cup A_3^c} t_{ji}x_{ji}^v+\sum_{k\in K}\sum_{i\in N^c:(O(k),i)\in A_1^c\cup A_3^c}\sigma(k)\gamma_{ik}^{vr}x_{O(k)i}^v\\&+\sum_{k\in K}\sum_{i\in N^c:(i,D(k))\in A_1^c\cup A_3^c}\left(-\tau(k)+t_{iD(k)}\right)\theta_{iD(k)}^{vr}x_{iD(k)}^v-T\sum_{(i,j)\in A_1^c\cup A_3^c}\lambda_{ij}^{vr}x_{ij}^v\leq 0\quad\forall v\in V.\end{aligned}\tag{4.58}$$

If [DSPvt] is optimal we obtain the feasible solution $(\hat{\pi}_i^v, \hat{\theta}_{ik}^v, \hat{\gamma}_{ik}^v, \hat{\lambda}_{ij}^v)$, and add optimality cuts (4.59) to Ω^{opt}:

$$\begin{aligned}Z^v\geq&\sum_{i\in N^c}\hat{\pi}_i^{v}\sum_{j:(j,i)\in A_1^c\cup A_3^c} t_{ji}x_{ji}^v+\sum_{k\in K}\sum_{i\in N^c:(O(k),i)\in A_1^c\cup A_3^c}\sigma(k)\hat{\gamma}_{ik}^{v}x_{O(k)i}^v\\&+\sum_{k\in K}\sum_{i\in N^c:(i,D(k))\in A_1^c\cup A_3^c}\left(-\tau(k)+t_{iD(k)}\right)\hat{\theta}_{iD(k)}^{v}x_{iD(k)}^v-T\sum_{(i,j)\in A_1^c\cup A_3^c}\hat{\lambda}_{ij}^{v}x_{ij}^v\quad\forall v\in V.\end{aligned}\tag{4.59}$$

As all vehicles are identical, any cut obtained from [DSPvt] is added for all vehicles $v \in V$, to speed up the solution procedure. We repeat the decomposition procedure until [RMPt] gives the same objective bound as the feasible solution found in [DSPvt], indicating the optimal solution to [CM2].

4.3 Relaxing time and commodity constriants

The third option to decompose [CM2] is to relax both commodity and time constriants. In such a case [MPyt] gives a set of vehicle movements that may not be feasible in terms of time constraints or commodity flow constraints. The [MPyt] is reformulated as:

[MPyt]

$$\min \quad \sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V} c_{ij}x_{ij}^v + \sum_{v\in V} Z^v \tag{4.60}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A_1^c\cup A_3^c} x_{ij}^v - \sum_{j:(j,i)\in A_1^c\cup A_3^c} x_{ji}^v = 0 \qquad \forall i\in N^c, \forall v\in V, \tag{4.61}$$

$$\sum_{(i,j)\in A_1^c\cup A_3^c : i\in N_s^c} x_{ij}^v \leq 1 \qquad \forall v\in V, \tag{4.62}$$

$$\sum_{(i,j)\in A_1^c\cup A_3^c}\sum_{v\in V} \Theta_{ij}^{r^y} x_{ij}^v \geq \Theta_0^{r^y} \qquad \forall r^y \in \Omega_y^{\text{feas}}, \tag{4.63}$$

$$z^v + \sum_{(i,j)\in A_1^c\cup A_3^c} \Lambda_{ij}^{vh} x_{ij}^v \geq \Lambda_0^{vh} \qquad \forall v\in V, \forall h\in\Omega^{\text{opt}}, \tag{4.64}$$

$$\sum_{(i,j)\in A_1^c\cup A_3^c} \Theta_{ij}^{vr^t} x_{ij}^v \geq \Theta_0^{vr^t} \qquad \forall v\in V, \forall r^t\in\Omega_t^{\text{feas}}, \tag{4.65}$$

$$x_{ij}^v \quad \text{binary} \qquad \forall (i,j)\in A_1^c\cup A_3^c, \forall v\in V. \tag{4.66}$$

We solve a relaxed [MPyt] on subsets $\bar{\Omega}_y^{\text{feas}} \subset \Omega_y^{\text{feas}}$, $\bar{\Omega}_t^{\text{feas}} \subset \Omega_t^{\text{feas}}$, $\bar{\Omega}_y^{\text{opt}} \subset \Omega_t^{\text{opt}}$. In each iteration [DSPy] and [DSPvt] are solved, and if necessary, cuts (4.29), (4.58), and(4.59) are added to $\bar{\Omega}_y^{\text{feas}}, \bar{\Omega}_t^{\text{feas}}, \Omega^{\text{opt}}$. The procedure continues until the solution of [RMPyt] is feasible for [DSPy], and gives the same objective bound as the feasible solution found in [DSPvt], indicating the optimal solution to [CM2]. The complete Benders decomposition algorithm is given below.

Complete Benders decomposition algorithm.

Step 1:

Solve [RMPyt] and obtain $\hat{x}_{ij}^{v}$.

Step 2:

Solve [DSPy], and [DSPvt] at $\hat{x}_{ij}^{v}$.

- If both [DSPy] and [DSPvt] are feasible, and value of variables $z^v, v \in V$ are equal to their corresponding objectives in [DSPvt], $\hat{x}_{ij}^{v}$ is the optimal solution to [CM2]. Stop.

- If [DSPy] is unbounded generate cuts (4.29) and update $\bar{\Omega}_{y}^{\text{feas}}$.

- If [DSPvt] is unbounded generate cuts (4.58) and update $\bar{\Omega}_{t}^{\text{feas}}$.

- Else if [DSPvt] is feasible generate cuts (4.59) and update $\bar{\Omega}_{t}^{\text{opt}}$.

- Go to Step 1.

We analyze the performance of each decomposition approach in Section 6.2. Results show that applying Benders decomposition to [CM2] does not produce desirable results. The rate of convergence to the optimal solution is poor and requires a high computational effort. To improve the convergence rate and reduce the computational effort, we develop several algorithm improvements, presented in Chapter 5.

Chapter 5

Algorithm improvements

Solving [CM2] by Benders decomposition turned out to be not as promising as first expected. The procedure requires a high amount of CPU time to solve and the convergence rate is very poor. In fact applying Benders decomposition to even small sized problems did not solve the model by any of the Benders decomposition approaches discussed in Chapter 4. The major weakness is that the subproblem is a feasibility problem. This is known to have convergence issues in Benders decomposition (Codato and Fischetti, 2006). To overcome this issue we develop a reduction technique, a family of feasibility cuts to tighten the relaxed master problem, and employ a multi-cut Benders decomposition approach.

5.1 Reducing [SPvt] to a feasibility problem

In this section we present a Lemma that reduces subproblem [SPvt] based on the characteristic of the SNDP. Currently subproblem [SPvt] verifies the feasibility and optimality of the solution from the relaxed master problems [RMPt] or [RMPyt]. However it can be shown that this problem could be transformed into a feasibility subproblem, if waiting time cost c_i of vehicles is the same in all terminals. This is shown by Lemma 2:

Lemma 2. *Given $c_i = c, \forall i \in N^c$, the objective function (3.22) can be written as:*
$$\sum_{(i,j)\in A_1^c \cup A_3^c} \sum_{v\in V} c_{ij}x_{ij}^v + \sum_{i\in N^c}\sum_{v\in V} c_i \left(\sum_{j\in N^c} w_{ij}^v - \sum_{j\in N^c} \left(w_{ji}^v + t_{ji}x_{ji}^v\right) \right) = \sum_{(i,j)\in A_1^c \cup A_3^c} \sum_{v\in V} (c_{ij} - ct_{ij})x_{ij}^v.$$

Proof. For $x_{ij}^v = 0$, we have by equation (3.19), $w_{ij}^v = 0$, and by equation (3.16), $w_{ji}^v = 0$ leading to a value of zero in both sides of the above equation. Let
$\digamma^v = (i_1, i_2, ..., i_{n-1}, i_n, i_{n+1}, ..., i_\kappa, i_1)$ denote the cycle of vehicle $v \in V$, we have
$x_{i_n i_{n+1}} = 1, \forall i \in N^c, i_n \in \digamma^v$, and:
$$\sum_{j\in N^c} w_{ij}^v - \sum_{j\in N^c} \left(w_{ji}^v + t_{ji}\right)$$
$$= w_{n_1n_2} - (w_{n_2n_3} + t_{n_2n_3}) + ... + w_{n_{i-1}n_i} - (w_{n_in_{i+1}} + t_{n_in_{i+1}}) + w_{n_in_{i+1}} - (w_{n_{i+1}n_{i+2}} + t_{n_{i+1}n_{i+2}})$$
$$+ w_{n_{i+1}ni+2} - (w_{n_{i+2}n_{i+3}} + t_{n_{i+2}n_{i+3}}) + ... + w_{n_\kappa n_1} - (w_{n_1n_2} + t_{n_1n_2})$$
$$= -t_{n_2n_3} - ... - t_{n_in_{i+1}} - t_{n_{i+1}n_{i+2}} - t_{n_{i+2}n_{i+3}} - ... - t_{n_1n_2}$$
$$= - \sum_{(j,i)\in A_1^c \cup A_3^c : (n_j, n_i)\in \digamma^v} t_{ji}$$
$$= - \sum_{(i,j)\in A_1^c \cup A_3^c : (n_i, n_j)\in \digamma^v} t_{ij}. \qquad \square$$

To further clarify we have $t_{ij} = -T + t_{D(k)O(k')}, i = D'(k) \in N_S^c, k, k' \in K$. Therefore, the objective function may be written as: $\sum_{(i,j)\in A_1^c \cup A_3^c} \sum_{v\in V} (c_{ij} - ct_{ij})x_{ij}^v = \sum_{(i,j)\in A_1^c \cup A_3^c} c_{ij}x_{ij}^v + c(T - \sum_{(i,j)\in A_1^c} t_{ij} + t_{D(k)D'(k)} + t_{D'(k)O(k')})$, corresponding to the total waiting time of vehicle v. Note that Lemma 2 holds for vehicle specific waiting time costs c^v, however, all vehicles are identical in our problem.

Lemma 2 shows that if $c_i = c, \forall i \in N^c$, vehicle waiting cost is independent of variables w_{ij}^v, and objective function 3.22 in [CM2] may be reformulated. This assumption holds for SNDP as waiting cost of vehicles is independent of the terminal. As a consequence to Lemma 2 only feasibility cuts (4.58) are added to Ω_t^{feas} in the Benders decomposition algorithm, and the first feasible solution to all subproblems is the optimal solution to [CM2].

5.2 Valid cuts for SNDP

This section presents a family of valid cuts that are valid for SNDP, and are used to tighten the relaxed Benders master problems. These cuts are redundant in [CM2], but become active in the relaxed master problems. By exploiting the characteristics of a feasible cycle in the continuous time model, we can introduce cuts aiming to enforce a relationship between selected nodes served by any vehicle. These cuts greatly affect the starting lower bound given by the solution of the relaxed master problems, and also reduce the number of required iterations for the algorithms to converge. However they incur a higher computational burden on solving the master problems in each iteration. A balance must be struck between which cuts to include in the master problems to enable a reasonable solution time, while maintaining a desirable quality of the solution lower bound. The family of feasibility cuts is presented in the following sections

5.2.1 Covering cuts

The covering cuts (5.1)-(5.4) enforce that all origins and destinations are visited by a minimum of $\lceil d_k/\eta \rceil$ vehicles. Theorem 1 states that a feasible solution of [CM2] may not require vehicles to move through all origins $O(k), k \in K$ (or destinations $D(k), k \in K$), and some commodities $k' \in K, k \neq k$ may be picked up (dropped off) at nodes other than their origin $O(k')$ (destination $D(k')$). Constraints (5.1), ((5.2)) enforce a vehicle to move through at least one of these origins or destinations. This setting allows the simultaneous pick up or delivery of Theorem 1.

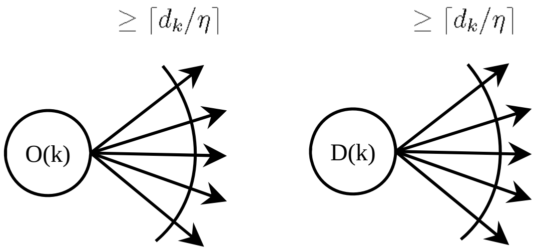
For origins $O(k)$, such that $((O(k), j') \notin A_2^c, \forall j' \in N^c)$ we add cuts (5.1) to the master problem. For destinations $D(k)$ such that $((j', D(k)) \notin A_2^c, \forall j' \in N^c)$ we add cuts (5.2) to the master problem.

$$\sum_{v \in V} \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c} x_{jO(k)}^v \geq \lceil d_k/\eta \rceil \qquad \forall k \in K : (O(k), j') \notin A_2^c, \forall j' \in N^c, \quad (5.1)$$

$$\sum_{v \in V} \sum_{j:(j,D(k)) \in A_1^c \cup A_3^c} x_{jD(k)}^v \geq \lceil d_k/\eta \rceil \qquad \forall k \in K : (j', D(k)) \notin A_2^c, \forall j' \in N^c. \quad (5.2)$$

Constraints (5.1) force a minimum $\lceil d_k/\eta \rceil$ of vehicles $v \in V$ to enter origin $O(k)$. Similarly Constraints (5.2) force a minimum $\lceil d_k/\eta \rceil$ of vehicles $v \in V$ to enter destination $D(k)$. Figure 5.1 shows an example of cuts 5.1, and 5.2.

Figure 5.1: Covering cuts 5.1, and 5.2.

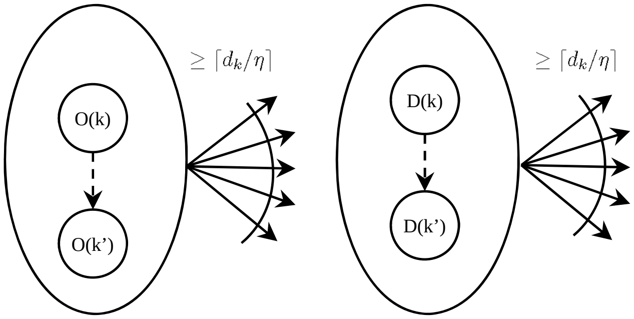


For the rest of the commodities $k \in K$ that can be picked up in nodes other than their origin $O(k)$, we add cuts (5.3) to the master problems. For commodities $k \in K$ that can be delivered to nodes other than their destination $D(k)$, we add cuts (5.4) to the master problem.

$$\sum_{v\in V}\sum_{j:(j,O(k))\in A_1^c\cup A_3^c} x^v_{jO(k)} + \sum_{v\in V}\sum_{i:(O(k),i)\in A_2^c}\sum_{j:(i,j)\in A_1^c\cup A_3^c} x^v_{ij} \geq \lceil d_k/\eta \rceil$$
$$\forall k\in K : \exists j' \in N^c|(O(k),j')\in A_2^c, \quad (5.3)$$
$$\sum_{v\in V}\sum_{j:(j,D(k))\in A_1^c\cup A_3^c} x^v_{jD(k)} + \sum_{v\in V}\sum_{i:(i,D(k))\in A_2^c}\sum_{j:(i,j)\in A_1^c\cup A_3^c} x^v_{ij} \geq \lceil d_k/\eta \rceil$$
$$\forall k\in K : \exists j' \in N^c|(j',D(k))\in A_2^c. \quad (5.4)$$

Constraints (5.3) force a minimum $\lceil d_k/\eta \rceil$ of vehicles $v \in V$ to either enter origin $O(k)$ or a node $i \in N^c$ such that $(O(k), i) \in A_2^c$, where commodity k may be picked up, without requiring vehicle movement to $O(k)$. Constraints (5.4) force a minimum $\lceil d_k/\eta \rceil$ of vehicles $v \in V$ to either enter destination $D(k)$ or a node $i \in N^c$ such that $(i, D(k)) \in A_2^c$, where commodity k may be dropped off, without requiring vehicle movement to $D(k)$. Figure 5.2 shows an example of covering cuts 5.3, and 5.4.

Figure 5.2: Covering cuts 5.3, and 5.4.



5.2.2 Origin-Destination cuts

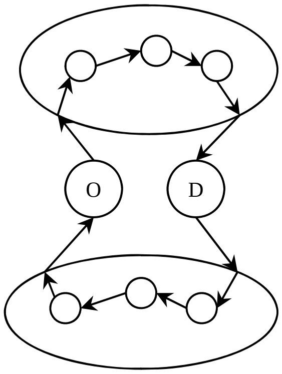
The Origin-Destination cuts ensure that when a vehicle $v \in V$ is assigned to service a commodity $k \in K$, it picks it up and delivers it. According to Theorem 1, a commodity $k \in K$ may be picked up (dropped off) without a visit to $O(k)$ ($D(k)$) if there is an arc $(O(k), i) \in A_2^c$ ($(i, D(k)) \in A_2^c$). Cuts are written depending on whether these arcs exist.

For commodities $k \in K$ such that $((O(k), j') \notin A_2^c, \forall j \in N^c)$ and $((j, D(k)) \notin A_2^c, \forall j \in N^c)$, we add cuts (5.5).

$$\sum_{j:(O(k),j) \in A_1^c \cup A_3^c} x^v_{O(k)j} = \sum_{j:(D(k),j) \in A_1^c \cup A_3^c} x^v_{D(k)j} \qquad \forall v \in V, \forall k \in K : (O(k), j') \notin A_2^c, \forall j' \in N^c, \\ (j', D(k)) \notin A_2^c, \forall j' \in N^c. \quad (5.5)$$

Cuts (5.5) enforce the visiting vehicle v to origin $O(k)$ to visit $D(k)$. Figure 5.3 shows an example of such cuts.

Figure 5.3: Origin-Destination cuts 5.5.



For commodities $k \in K$ such that $((O(k), j) \notin A_2^c, \forall j \in N^c)$, but $\exists j \in N^c | (j', D(k)) \in$

A_2^c we add cuts (5.6), and (5.7):

$$\sum_{j:(D(k),j)\in A_1^c\cup A_3^c} x_{D(k)j}^v \leq \sum_{j:(O(k),j)\in A_1^c\cup A_3^c} x_{O(k)j}^v \qquad \forall v \in V, \forall k \in K : \exists j' \in N^c|(j', D(k)) \in A_2^c, (O(k), j'') \notin A_2^c, \forall j'' \in N^c, \quad (5.6)$$

$$\sum_{j:(O(k),j)\in A_1^c\cup A_3^c} x_{O(k)j}^v \leq \sum_{i:(i,D(k))\in A_2^c} \sum_{j:(i,j)\in A_1^c\cup A_3^c} x_{ij}^v + \sum_{j:(D(k),j)\in A_1^c\cup A_3^c} x_{D(k)j}^v$$
$$\forall v \in V, \forall k \in K : \exists j' \in N^c|(j', D(k)) \in A_2^c, (O(k), j'') \notin A_2^c, \forall j'' \in N^c. \quad (5.7)$$

Cuts (5.6) ensure that if destination $D(k)$ is visited by a vehicle $v \in V$, it visits origin $O(k)$. If origin $O(k)$ is however visited, by cut (5.7) we force vehicle v to visit either $D(k)$ or a node i such that arc $(i, D(k)) \in A_2^c$ exists, and where commodity k may be dropped off.

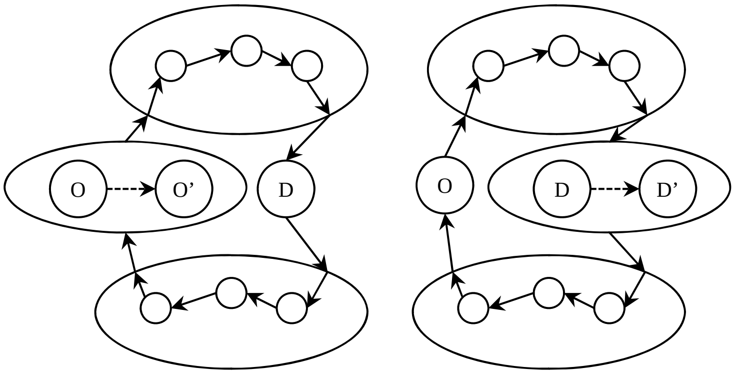
For commodities $k \in K$ such that $(j, D(k)) \notin A_2^c, \forall j \in N^c$, but $\exists j \in N^c|(O(k), j) \in A_2^c$, we add cuts (5.8) and (5.9):

$$\sum_{j:(O(k),j)\in A_1^c\cup A_3^c} x_{O(k)j}^v \leq \sum_{j:(D(k),j)\in A_1^c\cup A_3^c} x_{D(k)j}^v \qquad \forall v \in V, \forall k \in K : \exists j' \in N^c|(O(k), j') \in A_2^c, (j'', D(k)) \notin A_2^c, \forall j'' \in N^c, \quad (5.8)$$

$$\sum_{j:(D(k),j)\in A_1^c\cup A_3^c} x_{D(k)j}^v \leq \sum_{i:(O(k),i)\in A_2^c} \sum_{j:(i,j)\in A_1^c\cup A_3^c} x_{ij}^v + \sum_{j:(O(k),j)\in A_1^c\cup A_3^c} x_{O(k)j}^v$$
$$\forall v \in V, \forall k \in K : \exists j' \in N^c|(O(k), j') \in A_2^c, (j'', D(k)) \notin A_2^c, \forall j'' \in N^c. \quad (5.9)$$

Cuts (5.8) ensure that if origin $O(k)$ is visited by a vehicle $v \in V$, it visits destination $D(k)$. If destination $D(k)$ is however visited, by cut (5.9) we force vehicle v to visit either $O(k)$ or a node i such that arc $(O(k), i) \in A_2^c$ exists, and commodity k may be picked up there. Figure 5.4 gives an example of Origin-Destination cuts 5.6-5.9.

Figure 5.4: Origin-Destination cuts 5.6-5.9.



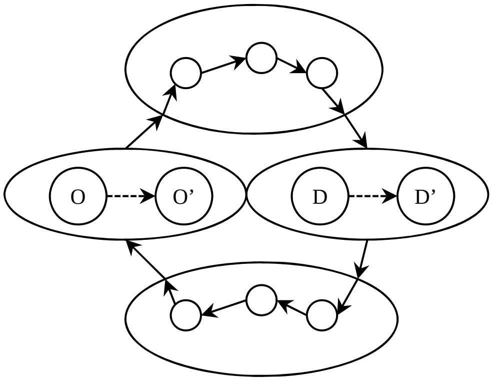
And finally for commodities $k \in K$ such that $\exists j \in N^c | (j, D(k)) \in A_2^c$ and $\exists j \in N^c | (O(k), j) \in A_2^c$ we add cuts (5.10) and (5.11):

$$\sum_{j:(O(k),j)\in A_1^c \cup A_3^c} x^v_{O(k)j} \leq \sum_{i:(i,D(k))\in A_2^c} \sum_{j:(i,j)\in A_1^c \cup A_3^c} x^v_{ij} + \sum_{j:(D(k),j)\in A_1^c \cup A_3^c} x^v_{D(k)j}$$
$$\forall v \in V, \forall k \in K : \exists j' \in N^c | (j', D(k)) \in A_2^c, \exists j' \in N^c | (O(k), j') \in A_2^c, \quad (5.10)$$

$$\sum_{j:(D(k),j)\in A_1^c \cup A_3^c} x^v_{D(k)j} \leq \sum_{i:(O(k),i)\in A_2^c} \sum_{j:(i,j)\in A_1^c \cup A_3^c} x^v_{ij} + \sum_{j:(O(k),j)\in A_1^c \cup A_3^c} x^v_{O(k)j}$$
$$\forall v \in V, \forall k \in K : \exists j' \in N^c | (j', D(k)) \in A_2^c, \exists j' \in N^c | (O(k), j') \in A_2^c. \quad (5.11)$$

Cuts (5.10) ensure that if origin $O(k)$ is visited by a vehicle $v \in V$, it either visits $D(k)$ or a node i such that arc $(i, D(k)) \in A_2^c$ exists, and where commodity k may be dropped off. Cuts (5.11) ensure that if destination $D(k)$ is visited by a vehicle $v \in V$, it either visits $O(k)$ or a node i such that arc $(O(k), i) \in A_2^c$ exists, and where commodity k may be dropped off. Figure 5.5 gives an example of such cuts.

Figure 5.5: Origin-Destination cuts 5.10 and 5.11.



5.2.3 Subtour elimination cuts and precedence cuts

This section presents a set of modified Miller-Tucker-Zemlin (MTZ) (Miller et al., 1960) cuts that are used in SNDP for three purposes. First, when relaxing time constraints, master problems [RMPt] and [RMPyt] may result in subtours. In this case, a vehicle cycle does not visit an end of horizon node $i \in N_s^c$, or would move on two or more independent cycles. We later show in section 5.2.6 how we can force the vehicles used to move through end of horizon nodes. However, we may still reach results with more than one independent cycle for a vehicle as shown by Figure 5.6. We therefore employ MTZ constraints used for the very purpose of subtour elimination in the literature. Let μ_i^v denote the placement of node $i \in N^c$ in a cycle of vehicle $v \in V$. By introducing cut (5.12) and (5.13) we remove the possibility of subtours in the solution to the relaxed master problems.

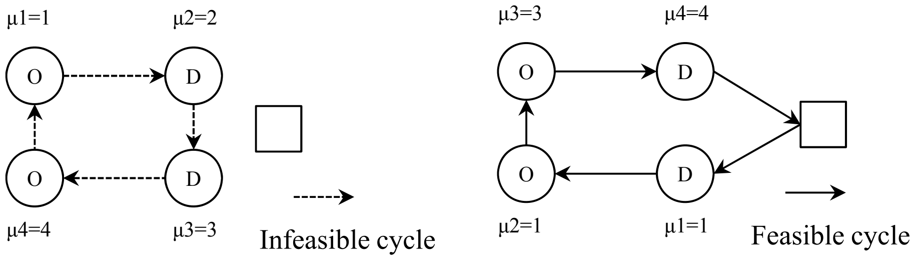
$$\mu_i^v - \mu_j^v + 1 \leq (|N^c| - 1)(1 - x_{ij}^v) \qquad \forall (i,j) \in A_1^c, \forall v \in V, \quad (5.12)$$

$$\mu_1 = 1, \quad (5.13)$$

$$\mu_i^v \geq 0 \qquad \forall i \in N^c, \forall v \in V. \quad (5.14)$$

Covering cuts (5.1)-(5.4), and Origin-Destination cuts (5.5)-(5.11) ensure that com-

Figure 5.6: Infeasible cycle due to subtour, and a feasible cycle.



modities $k \in K$ are picked up and delivered by the same vehicle $v \in V$, but do not force commodity k to be picked up before it is delivered. Master problem solutions may contain cycles that $D(k)$ is visited before $O(k)$. Such cycles are infeasible to [CM2]. To eliminate the infeasible cycles we extend the MTZ constraints so that $O(k)$ precedes $D(k)$:

$$\mu_i^v - \mu_j^v + 1 \le (|N^c| - 1)(1 - x_{ij}^v) \qquad \forall (i,j) \in A_1^c, \forall v \in V \tag{5.15}$$

$$\mu_{O(k)}^v \le \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c, j \in N_s^c} x_{jO(k)}^v + (|N^c| - 1)\left(1 - \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c, j \in N_s^c} x_{jO(k)}^v\right)$$
$$\forall k \in K, \forall v \in V, \tag{5.16}$$

$$\mu_{O(k)}^v \ge \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c, j \in N_s^c} x_{jO(k)}^v \qquad \forall k \in K, \forall v \in V, \tag{5.17}$$

$$\mu_{O(k)}^v \le \mu_{D(k)}^v \qquad \forall k \in K, \forall v \in V, \tag{5.18}$$

$$\mu_{D(k)}^v \ge \mu_i^v - (|N^c| - 1)\left(\sum_{j:(O(k),j) \in A_1^c \cup A_3^c} x_{O(k)j}^v + \sum_{i':(O(k),i') \in A_2^c, i' \neq i} \sum_{j:(i',j) \in A_1^c \cup A_3^c} x_{i'j}^v\right)$$
$$\forall i : (O(k), i) \in A_2^c, \forall k \in K, \forall v \in V, \tag{5.19}$$

$$\mu_{O(k)}^v \le \mu_i^v + (|N^c| - 1)\left(\sum_{j:(D(k),j) \in A_1^c \cup A_3^c} x_{D(k)j}^v + \sum_{i':(i',D(k)) \in A_2^c, i' \neq i} \sum_{j:(i',j) \in A_1^c \cup A_3^c} x_{i'j}^v\right)$$
$$\forall i : (i, D(k)) \in A_2^c, \forall k \in K, \forall v \in V, \tag{5.20}$$

$$\mu_i^v \ge 0 \qquad \forall i \in N^c, \forall v \in V. \tag{5.21}$$

Cuts (5.15) set the sequence of nodes in a vehicle cycle $v \in V$. Cuts (5.16) and (5.17) together set the sequence of the first visited node in a cycle (first node visited after the end of horizon node) to 1. Cuts (5.18) set the sequence of $O(k), k \in K$ to precede the sequence of $D(k)$. Constraints (5.19) cover cases where a commodity k is picked up at another node i such that $\exists i \in N^c | (O(k), i) \in A_2^c$. For these cases we set the sequence of $D(k)$ to proceed the sequence of i unless, $O(k)$, or another node i', such that $\exists i' \in N^c | (O(k), i') \in A_2^c$ is visited by vehicle v. Similarly cuts (5.20) cover cases where a commodity k is dropped off at another node i such that $\exists i \in N^c | (i, D(k)) \in A_2^c$. For these cases we set the sequence of $O(k)$ to precede the sequence of i unless, $D(k)$, or another node i', such that $\exists i' \in N^c | (i', D(k)) \in A_2^c$ is visited by vehicle v.

The MTZ constraints may be extended to model time constraints. By modifying the definition of μ_i^v to be the arrival time of vehicle $v \in V$ at node $i \in N^c$, we can enforce the time constraints of all commodities $k \in K$. In other words the modified MTZ constraints can replace the original time constraints (3.29)-(3.32). However, we do not replace the original time constraint in [CM2] by the modified MTZ constraints, as the definition of μ_i^v does not provide a linear approach to calculating the vehicle waiting costs in the objective function. Lemma 2 shows that vehicle waiting cost may be solely formulated based on variables x_{ij}^v. We therefore replace the original time constraints in the Benders relaxed master problems by the modified MTZ constraints (5.22)-(5.28):

$$\mu_i^v - \mu_j^v + t_{ij} \leq T(1 - x_{ij}^v) \qquad \forall (i,j) \in A_1^c, \forall v \in V \tag{5.22}$$

$$\mu_{O(k)}^v \leq \sigma(k) \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c, j \in N_s^c} x_{jO(k)}^v + T \left(1 - \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c, j \in N_s^c} x_{jO(k)}^v \right) \qquad \forall k \in K, \forall v \in V, \tag{5.23}$$

$$\mu_{O(k)}^v \geq \sigma(k) \sum_{j:(j,O(k)) \in A_1^c \cup A_3^c} x_{jO(k)}^v \qquad \forall k \in K, \forall v \in V, \tag{5.24}$$

$$\mu_{D(k)}^v \leq \tau(k) \qquad \forall k \in K, \forall v \in V, \tag{5.25}$$

$$\mu_{O(k)}^v \leq \mu_{D(k)}^v \qquad \forall k \in K, \forall v \in V, \tag{5.26}$$

$$\mu_{D(k)}^v \geq \mu_i^v - T \left(\sum_{j:(O(k),j) \in A_1^c \cup A_3^c} x_{O(k)j}^v + \sum_{i':(O(k),i') \in A_2^c, i' \neq i} \sum_{j:(i',j) \in A_1^c \cup A_3^c} x_{i'j}^v \right) \qquad \forall i : (O(k), i) \in A_2^c, \forall k \in K, \forall v \in V, \tag{5.27}$$

$$\mu_{O(k)}^v \leq \mu_i^v + T \left(\sum_{j:(D(k),j) \in A_1^c \cup A_3^c} x_{D(k)j}^v + \sum_{i':(i',D(k)) \in A_2^c, i' \neq i} \sum_{j:(i',j) \in A_1^c \cup A_3^c} x_{i'j}^v \right) \qquad \forall i : (i, D(k)) \in A_2^c, \forall k \in K, \forall v \in V, \tag{5.28}$$

$$\mu_i^v \geq 0 \qquad \forall i \in N^c, \forall v \in V. \tag{5.29}$$

Cuts (5.22) enable the model to track transportation time in the cycle visited by a vehicle $v \in V$. Cuts (5.23) together with cuts (5.24) set the arrival time of the first visited node of all cycles (first node visited after the end of horizon node) to its availability time $\sigma(k)$. For all other origins $O(k'), k' \in K, k' \neq k$, cuts (5.24) set the arrival time to at least $\sigma(k')$. Cuts (5.25) limit the arrival of all destinations $D(k), k \in K$, to their delivery deadline $\tau(k)$. Cuts (5.26) serve as precedence constraints and set the arrival time of origins $O(k)$ to precede the arrival time of their destinations $D(k)$. Cuts (5.27) and (5.28) enable precedence consideration for commodities picked up or delivered from nodes other than their origin $O(k)$ and destination $D(k)$. Although the modified MTZ constraints are very useful in enabling a time feasible solution, they greatly increase the computational burden of solving large instances. The increase of burden is however lower than the original timing constraints of the relaxed master problem [RMPy], as shown by the numerical results in Chapter 6.

5.2.4 Vehicle-Commodity cuts

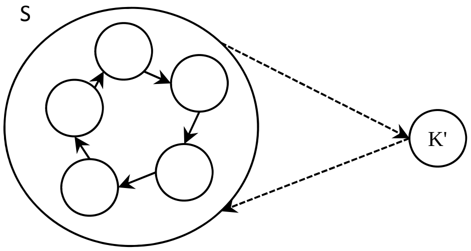
Vehicle-Commodity cuts are defined as commodities that cannot be served by the same vehicle. A vehicle that services such commodities violates the timing constraints and cannot deliver the commodities before their delivery deadline $\tau(k)$. Let $S \subset K$ be a subset of commodities that may be serviced by the same vehicle and let $\bar{S} \subset K$ be the set of all commodities $k' \in K$ such that commodities in $S \cup \{k'\}$ cannot be serviced by the same vehicle; i.e. the vehicle carrying commodities $S \cup \{k'\}$ violates the delivery deadline of at least one commodity in set $S \cup \{k'\}$. Hence, a cut that excludes commodities of set $S \cup \{k'\}$ from being assigned to the same vehicle is a valid cut for [CM2], and the relaxed master problems. Cuts (5.30), (5.31) are introduced for that purpose:

$$\sum_{i:(i,j)\in A_1^c \cup A_3^c} \sum_{j:j=O(k),k\in\bar{S}} x_{ij}^v + \sum_{i:(i,j)\in A_1^c \cup A_3^c} \sum_{j:j=D(k),k\in\bar{S}} x_{ij}^v \leq 2|\bar{S}| \left(|S| - \sum_{i:(i,j)\in A_1^c \cup A_3^c} \sum_{j:j=O(k),k\in S} x_{ij}^v \right) \quad \forall v \in V, \tag{5.30}$$

$$\sum_{i:(i,j)\in A_1^c \cup A_3^c} \sum_{j:j=O(k),k\in\bar{S}} x_{ij}^v + \sum_{i:(i,j)\in A_1^c \cup A_3^c} \sum_{j:j=D(k),k\in\bar{S}} x_{ij}^v \leq 2|\bar{S}| \left(|S| - \sum_{i:(i,j)\in A_1^c \cup A_3^c} \sum_{j:j=D(k),k\in S} x_{ij}^v \right) \quad \forall v \in V. \tag{5.31}$$

Cuts (5.30) restrict a vehicle $v \in V$ visiting the origins $O(k)$ of all commodities $k \in S$ from visiting any of the origins $O(k')$, or destinations $D(k')$ of commodities $k' \in \bar{S}$. The cut is only active if vehicle v visits all origins $O(k), k \in S$, and becomes redundant if the origin of at least one commodity $k \in S$ is not visited. Similarly cuts (5.31) restrict a vehicle $v \in V$ visiting the origins $D(k)$ of all commodities $k \in S$ from visiting any of the origins $O(k')$, or destinations $D(k')$ of commodities $k' \in \bar{S}$. The cut is only active if vehicle v visits all destinations $D(k), k \in S$, and becomes redundant if the destination of at least one commodity $k \in S$ is not visited. Figure 5.7 shows an example where commodities of set S cannot be served together with commodity k'.

Figure 5.7: Exclusive commodity cuts.



There is an exponential number of sets $S \subset K$ (bounded by $2^{|K|}$). Generating all such subsets is not possible. Given a subset of commodities S, and for $|S| = 2$, one can verify whether the two commodities $k_1, k_2 \in S$ can be served by one vehicle, by checking the availability times $\sigma(k_1), \sigma(k_2)$, and delivery deadlines $\tau(k_1), \tau(k_2)$. There can be 6 possible paths including the origins and destination of k_1, k_2:
$(O(k_1), O(k_2), D(k_1), D(k_2))$, $(O(k_1), O(k_2), D(k_2), D(k_1))$, $(O(k_1), D(k_1), O(k_2), D(k_2))$, $(O(k_2), O(k_1), D(k_2), D(k_1))$, $(O(k_2), O(k_1), D(k_1), D(k_2))$, $(O(k_2), D(k_2), O(k_1), D(k_1))$. If any of these paths is feasible and satisfies the availability times $\sigma(k_1), \sigma(k_2)$, and delivery deadlines $\tau(k_1), \tau(k_2)$, then commodities k_1 and k_2 can be served by one vehicle. If all paths are infeasible then k_1 and k_2 must be served by different vehicles. For this case we set $S = \{k\}$, $\bar{S} = \{k'\}$, and add cuts (5.30), (5.31). Similarly we can set $S = \{k'\}$, $\bar{S} = \{k\}$

and add cuts (5.30), (5.31).

For $|S| = 3$, the exclusivity of commodities $k \in S$ may be verified using feasibility problem [RCM], which is a modification of [CM2] with $|V| = 1$, and solved on a restricted network $G = (N^{rc}, A^{rc})$ consisting of only node set $N^{rc} \subset N^c$, and arc set $A^{rc} \subset A^c$, corresponding to commodities $k \in S$.

[RCM]

$$\min \quad 0$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A_1^{rc}\cup A_3^{rc}} x_{ij} - \sum_{j:(j,i)\in A_1^{rc}\cup A_3^{rc}} x_{ji} = 0 \qquad \forall i \in N^{rc}, \quad (5.32)$$

$$\sum_{j:(i,j)\in A_1^{rc}\cup A_3^{c}rc} x_{ij} \leq 1 \qquad \forall i \in N^{rc}, \quad (5.33)$$

$$\sum_{k\in S} y_{ijk} \leq |S| x_{ij} \qquad \forall (i,j) \in A_1^{rc} \cup A_2^{rc}, \quad (5.34)$$

$$\sum_{j:(O(k),j)\in A_2^{rc}} y_{O(k)jk} = 1 \qquad \forall k \in S, \quad (5.35)$$

$$\sum_{j:(j,D(k))\in A_2^{rc}} y_{jD(k)k} = 1 \qquad \forall k \in S, \quad (5.36)$$

$$\sum_{j:(i,j)\in A_1^{rc}\vee(i,j)\in A_2^{rc},j=D(k)} y_{ijk} - \sum_{j:(j,i)\in A_1^{rc}\vee(j,i)\in A_2^{rc},j=O(k)} y_{jik} = 0$$
$$\forall i \in N^{rc} \setminus \{O(k), D(k)\}, \forall k \in S, \quad (5.37)$$

$$\sum_{j:(j,i)\in A_1^{rc}} (w_{ji} + t_{ji}x_{ji}) \leq \sum_{j:(i,j)\in A_1^{rc}\cup A_3^{rc}} w_{ij} \qquad \forall i \in N^{rc}, \quad (5.38)$$

$$w_{O(k)i} \geq \sigma(k) x_{O(k)i} \qquad \forall i \in N^{rc} : (O(k), i) \in A_1^{rc} \cup A_3^{rc}, \forall k \in S, \quad (5.39)$$

$$w_{iD(k)} + t_{iD(k)} x_{iD(k)} \leq \tau(k) x_{iD(k)} \qquad \forall i \in N^{rc} : (i, D(k)) \in A_1^{rc} \cup A_3^{rc}, \forall k \in S, \quad (5.40)$$

$$w_{ij} \leq T x_{ij} \qquad \forall (i,j) \in A_1^{rc} \cup A_3^{rc}, \quad (5.41)$$

$$x_{ij} \quad \text{binary}, \quad w_{ij} \geq 0 \qquad \forall (i,j) \in A_1^{rc}, \quad (5.42)$$

$$y_{ijk} \geq 0 \qquad \forall (i,j) \in A_2^{rc}, \forall k \in S. \quad (5.43)$$

Constraints (5.32), (5.33) ensure conservation of vehicle flow. Constraints (5.34) link commodity flow to vehicle flow. Constraints (5.35)-(5.37) enforce the vehicle to pick up and deliver commodities $k \in S$. Timing requirements are assured by Constraints (5.38)-(5.41).

The complete algorithm to generate Vehicle-Commodity cuts is given below.

Complete algorithm to generate Vehicle-Commodity cuts.

Initialize $|S| = 2$, maximum subset size $|S|_{max}$, and generate all sets $S \subset K$. For each set S do:

Step 1:

Set $\bar{S} = \emptyset$. While $|S| \leq |S|_{max}$ do:

Solve [RCM] for set S.

- If [RCM] is feasible go to step 2.

- Else if [RCM] is infeasible, we have $|S| = 2$, $S = \{k, k'\}$. Redefine set $S = \{k\}$, and set $\bar{S} = \{k'\}$. Generate cuts (5.30) and (5.31). Redefine set $S = \{k'\}$, and set $\bar{S} = \{k\}$. Generate cuts (5.30) and (5.31). Stop.

Step 2:

- For all commodities $k \in K \setminus S$, solve [RCM] for set $S \cup \{k\}$. If [RCM] is infeasible add k to $\bar{S}$. Generate cuts (5.30) and (5.31), for sets $S, \bar{S}$.

- Set $|S| = |S| + 1$, and generate all sets S by adding a commodity $k \in K \setminus S, k \notin \bar{S}$ to S. Go to Step 1.

In the current implementation we generate cuts (5.30), (5.31) for subsets of size $|S| = 1$. That is we verify exclusivity for $(|k| - 1)(|K| - 2)/2$ sets of size $|S|_{max} = 2$.

5.2.5 Direct capacity cuts

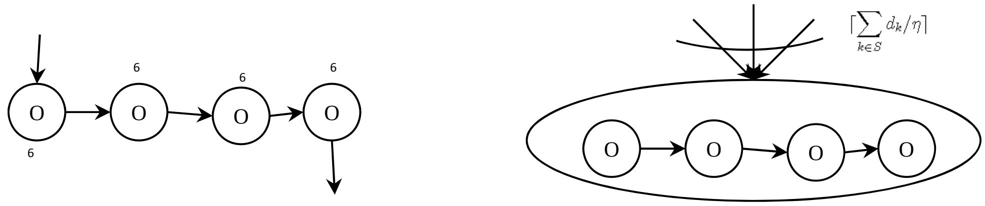
Direct capacity cuts prevent a consecutive pick up of commodities that violate vehicle capacity. Define set $S \subset K$ of $|S|$ commodities where $\sum\limits_{k \in S} d_k \geq \eta$. Commodities in set

S cannot be consecutively picked up by a single vehicle. Recall that if we visit an origin $O(k)$, we pick up all or part of its demand d_k. Therefore, if a vehicle consecutively visits all origins $O(k), k \in S$ then either it partially picks up at least one of the commodities $k \in S$, or the solution is infeasible. In other words, as a feasible solution satisfies vehicle capacity, then portion $\eta - \sum_{k \in S} d_k$ of the total demand must be picked up by another vehicle. For a set S we generate cut (5.44) and add it to the master problems:

$$\sum_{v' \neq v} \quad \sum_{j:(i,j) \in A_1^c \cup A_3^c, j \notin S} \quad \sum_{i \in S} x_{ij}^{v'} \geq \left(\sum_{(i,j) \in A_1^c \cup A_3^c : i,j \in S} x_{ij}^{v} - (|S| - 2) \right) \lceil \sum_{k \in S} d_k / \eta \rceil \quad \forall v \in V. \quad (5.44)$$

If a vehicle $v \in V$ consecutively visits all origins $O(k), k \in S$ then we have $\sum_{(i,j) \in A_1^c \cup A_3^c : i,j \in S} x_{ij}^{v} = |S| - 1$. In this case Cut (5.44) forces another vehicle $v' \in V, v' \neq v$ to visit at least one of the origins $O(k)$. Figure 5.8 gives an example with of a vehicle with capacity $\eta = 20$ consecutively visiting 4 origins, each with supply of 6. If a consecutive visit were to be made to these origins, at least $(3-2)(\lceil 24/20 \rceil) = 2$ vehicle are needed to visit these set of origins.

Figure 5.8: Direct capacity cuts.



a) Infeasible consecutive visit

b) Feasible consecutive visit

Set S is generated by choosing $|S|$ commodities $k \in K$ where $\sum_{k \in S} d_k \geq \eta$. In the current implementation we generate all sets of size $|S| = 3$, and add cuts (5.44) to the master problems.

5.2.6 Other cuts

This section presents a set of cuts that aid in preventing infeasible solutions of the master problems, or in speeding the decomposition process. These cuts are presented as follows.

- Maximum time cuts:

  These cuts restrict any cycle duration from exceeding the planning horizon T.

  $$\sum_{(i,j)\in A_1^c} t_{ij}x_{ij}^v \leq T \qquad \forall v \in V. \tag{5.45}$$

  Cut (5.45) require vehicle flow on arcs $(i,j) \in A_1^c$ to take less than T units of time.

- End of horizon cuts:

  These cuts enforce cycles to pass through an end of horizon node, as a feasible cycle contains one node $i \in N_S^c$.

  $$\sum_{(i,j)\in A_1^c} x_{ij}^v \leq |A_1^c| \sum_{(i,j)\in A_3^c} x_{ij} \qquad \forall v \in V. \tag{5.46}$$

  Cuts (5.46) enforce a vehicle $v \in V$ that flows on any arc $(i,j) \in A_1^c$ to pass through an arc $(i,j) \in A_3^c$.

- Vehicle preference cuts:

  All vehicles are identical in our problem. Vehicle preference cuts are introduced to enforce vehicle sequence and remove the redundancy in selecting vehicles from set V.

  $$\sum_{(i,j)\in A_1^c\cup A_3^c} x_{ij}^{v+1} \leq |A_1^c \cup A_3^c| \sum_{(i,j)\in A_1^c\cup A_3^c} x_{ij}^v \qquad \forall v \in V. \tag{5.47}$$

  Cuts 5.47 permit a vehicle $v+1$ to be used only if vehicle v is used.

5.3 Multi-cut Benders approach

The master problems of the continuous time SNDP are hard to solve. Our initial experiments showed that on average, 98% of the total CPU time in all Benders decomposition approaches is utilized by the commercial solver in solving the master problems. To increase the efficiency of Benders decomposition algorithm we aim to reduce the number of times we must solve the master problem. One approach to achieve this goal is the multi-cut Benders approach. In addition to the classical Benders approach where only feasibility and optimality cuts (4.29),(4.58), and (4.59) are added to the master problem, we seek to generate and add additional cuts that my lead to a reduction in the number of required iterations for the algorithm to converge. This section introduces a set of such cuts.

5.3.1 Disaggregated Commodity cuts

Subproblem [SPy] is an aggregated problem on the set of commodities $k \in K$. This subproblem is not separable by commodities $k \in K$ due to capacity constraints (3.25). Disaggregated commodity cuts may be derived, where the feasibility of each commodity route is sought. This is possible by relaxing constraint (4.12), and replacing it with an uncapacitated constraint (5.48), where only vehicle movement is enforced. This enables us to separate [SPy] into $|K|$ subproblems [SPky], one for each commodity $k \in K$:

[SPky]

$$\min \quad 0$$

$$\text{s.t.} \quad y^v_{ijk} \leq \hat{x}^v_{ij} \qquad \forall (i,j) \in A^c_2, \forall v \in, V \tag{5.48}$$

$$\sum_{v \in V} \sum_{j:(O(k),j) \in A^c_2} y^v_{O(k)jk} = 1, \tag{5.49}$$

$$\sum_{v \in V} \sum_{j:(j,D(k)) \in A^c_2} y^v_{jD(k)k} = 1, \tag{5.50}$$

$$\sum_{j:(i,j) \in A^c_1 \vee (i,j) \in A^c_2, j=D(k)} y^v_{ijk} - \sum_{j:(j,i) \in A^c_1 \vee (j,i) \in A^c_2, j=O(k)} y^v_{jik} = 0$$

$$\forall i \in N^c \setminus \{O(k), D(k)\}, \forall v \in V, \tag{5.51}$$

$$y^v_{ijk} \geq 0 \qquad \forall (i,j) \in A^c_2, \forall v \in V. \tag{5.52}$$

To derive Benders feasibility cuts for an infeasible solution $\hat{x}^v_{ij}$ in subproblem [SPky] we take the dual of [SPky]. Associating dual variables $\alpha^v_{ij}, \beta_{O(k)}, \beta_{D(k)}, \beta^v_{ik}$ with constraints (5.49)-(5.52) we have the dual of [SPky] as:

[DSPky]

$$\max \quad - \sum_{(i,j) \in A^c_1 \cup A^c_3} \sum_{v \in V} \hat{x}^v_{ij} \alpha^v_{ij} + \beta_{O(k)} + \beta_{D(k)} \tag{5.53}$$

$$\text{s.t.} \quad -\alpha^v_{O(k)j} + \beta_{O(k)} - \beta^v_{jk} \leq 0 \qquad \forall (O(k), j) \in A^c_1, j \neq D(k), \forall v \in V, \tag{5.54}$$

$$-\alpha^v_{O(k)D(k)} + \beta_{O(k)} - \beta_{D(k)} \leq 0 \qquad \forall (O(k), D(k)) \in A^c_1, \forall v \in V, \tag{5.55}$$

$$\beta_{O(k)} - \beta^v_{jk} \leq 0 \qquad \forall (O(k), j) \in A^c_2, \forall v \in V, \tag{5.56}$$

$$-\alpha^v_{D(k)j} + \beta_{D(k)} - \beta^v_{jk} \leq 0 \qquad \forall (D(k), j) \in A^c_1, \forall v \in V, \tag{5.57}$$

$$\beta_{D(k)} - \beta^v_{jk} \leq 0 \qquad \forall (D(k), j) \in A^c_2, \forall v \in V, \tag{5.58}$$

$$-\alpha^v_{ij} - \beta^v_{jk} \leq 0 \qquad \forall (i,j) \in A^c_1, i, j \notin \{O(k), D(k)\}, \forall v \in V, \tag{5.59}$$

$$-\alpha_{iO(k)}^{v} + \beta_{ik}^{v} - \beta_{O(k)} \leq 0 \qquad \forall (i, O(k)) \in A_1^c, \forall v \in V, \quad (5.60)$$

$$-\alpha_{iD(k)}^{v} + \beta_{ik}^{v} - \beta_{D(k)} \leq 0 \qquad \forall (i, D(k)) \in A_2^c, i \neq O(k), \forall v \in V, \quad (5.61)$$

$$\beta_{ik}^{v} - \beta_{O(k)} \leq 0 \qquad \forall (i, O(k)) \in A_2^c, \forall v \in V, \quad (5.62)$$

$$\beta_{ik}^{v} - \beta_{D(k)} \leq 0 \qquad \forall (i, D(k)) \in A_2^c, \forall v \in V, \quad (5.63)$$

$$\alpha_{ij}^{v} \geq 0 \qquad \forall (i, j) \in A_1^c, \forall v \in V. \quad (5.64)$$

Problem [DSPky] is solved at each iteration of the Benders algorithm under master problem solution $\hat{x}_{ij}^{v}$. The dual problem [DSPky] is always feasible as the all-zero solution is feasible. If [DSPky] is unbounded for a commodity $k \in K$, then the we derive the unbounded dual ray $(\alpha_{ij}^{vr}, \beta_{O(k)}^{r}, \beta_{D(k)}^{r})$, and add feasibility cut (5.65) to set Ω_{y}^{feas}.

$$-\sum_{(i,j) \in A_1^c \cup A_3^c} \sum_{v \in V} \alpha_{ij}^{vr} x_{ij}^{v} + \beta_{O(k)}^{r} + \beta_{D(k)}^{r} \leq 0. \quad (5.65)$$

5.3.2 Identical Vehicle cuts

Subproblems [SPy] and [SPky] are aggregated on vehicles $v \in V$. As multiple vehicles may satisfy the demand of a commodity $k \in K$ we cannot disaggregate these subproblems by vehicle. A solution $\hat{x}_{ij}^{v}$ of the master problem thus gives the movements of $|V^a| \leq |V|$ identical vehicles on network G^c, where $V^a \subseteq V$ denotes the set of active vehicles. Consider an example with $|V^a| = 2$. Let $\hat{x}_{ij}^{v} = [\hat{x}_{ij}^{1}, \hat{x}_{ij}^{2}]$ represent a solution of the master problem with $\hat{x}_{ij}^{1}$ indicating the movements of the first active vehicle, and $\hat{x}_{ij}^{1}$ the movements of the second active vehicle. Now note that although the solution $[\hat{x}_{ij}^{2}, \hat{x}_{ij}^{1}]$ pertains to the same solution to the overall problem it represents a different solution to the master problem. In other words even though the solutions represent the same overall vehicle movements, they do not represent the same solution of the master problem, due to vehicle sequence. If a Benders feasibility cut removes infeasible solution $[\hat{x}_{ij}^{1}, \hat{x}_{ij}^{2}]$, the infeasible solution $[\hat{x}_{ij}^{2}, \hat{x}_{ij}^{1}]$

may be generated in the next iterations. It is possible, in the worst case, that the master problem is solved $|V^a|$ times to remove the solutions pertaining to the same overall vehicle movements in the network.

To remove all such solutions at once, we derive cuts that remove any combination of a master problem solution by vehicle sequence. As all vehicles are sequenced by cuts (5.47) we can safely divide an unbounded dual ray α_{ij}^{vr} into $[\alpha_{ij}^{v^a r}, \alpha_{ij}^{v^{ina} r}]$, where $\alpha_{ij}^{v^a r}$ denotes the part of α_{ij}^{vr} corresponding to active vehicles, and $\alpha_{ij}^{v^{ina} r}$ to inactive vehicles. The vector $\alpha_{ij}^{v^a r}$ can further be separated into $|V^a|$ active vehicles $[\alpha_{ij}^{1^a r}, \alpha_{ij}^{2^a r}, ..., \alpha_{ij}^{v^a r}]$, which are used to generate all required cuts to remove the overall infeasible solution. We derive all possible combinations of $\alpha_{ij}^{v^a r}$ corresponding to the $|V^a|!$ possible placements of active vehicles. Each combination is then joined with $\alpha_{ij}^{v^{ina} r}$ to form a valid unbounded dual ray, and used to derive new infeasibility cuts (4.29) and (5.65).

5.3.3 Original network cuts

The continuous time model is defined on a network G^c. This network is built using the original network of terminals. Nodes in set N^c may be associated with the same terminal. A cycle $\digamma = (i_1, ..., i_n ..., i_\kappa, i_1)$ obtained from the solution of the master problems corresponds to a cycle $\digamma^o = (i_1^o, ..., i_n^o, ..., i_\kappa^o, i_1^o)$ in the original network. A cycle $\digamma^o$ in the original network may in turn correspond to a number of cycles in the continuous time network, any of which may or may not be feasible. All paths $\digamma$ corresponding to the original path $\digamma^o$ incur the same costs, as transportation costs c_{ij} and transportation times t_{ij} all correspond to the original network. Therefore it is possible that by removing an infeasible solution $\hat{x}_{ij}^v$ corresponding to the cycles $\digamma$ and $\digamma^o$, the master problem gives another solution with cycle $\digamma'$, corresponding to the same original cycle $\digamma^o$.

We can generate and remove cycles that correspond to an infeasible original cycle. Whenever the master problem gives an infeasible solution we derive its corresponding original cycle and use it to generate all possible cycles in network G^c. Infeasible cycles $\digamma$ are then removed using cut (5.66), generalized from the work of Ascheuer et al. (2000) for the travelling salesman problem.

$$\sum_{n:i_n \in \digamma, n \leq \kappa - 1} x^v_{i_n i_{n+1}} + x^v_{i_\kappa i_1} \leq \kappa - 1 = |\digamma| - 1 \qquad \forall v \in V. \quad (5.66)$$

Cuts (5.66) remove an infeasible cycle. These cuts may however be strengthened based on the source of a cycle's infeasibility. We assess a cycle to be infeasible depending on the reasons below:

1. The origin (destination) of a commodity is visited but the destination (origin) of that commodity is not visited.

   If the origin of a commodity $k \in K$ is visited but the destination of commodity k is not visited, cut (5.67) is a strengthening to (5.66), where m denotes the placement of $O(k)$ in cycle $\digamma$.

   $$\sum_{n:i_n \in \digamma, n \geq m, n \leq \kappa - 1} x^v_{i_n i_{n+1}} \leq \kappa - m - 1 \qquad \forall v \in V. \quad (5.67)$$

   If the destination of a commodity $k \in K$ is visited but the origin or a proxy origin to commodity k is not visited cut (5.68) is a strengthening to (5.66), where m denotes the placement of $D(K)$ in cycle $\digamma$.

   $$\sum_{n:i_n \in \digamma, n \leq m - 1} x^v_{i_n i_{n+1}} + x^v_{i_\kappa i_1} \leq m - 1 \qquad \forall v \in V. \quad (5.68)$$

2. The origin (destination) of a commodity $k \in K$ is visited after the destination (origin) of that commodity.

In this case the cut (5.69) is a strengthening of cut (5.66), where m and m' denote the placement of the origin and destination of commodity k.

$$\sum_{n:i_n \in F, n \geq m, n \leq m'-1} x^v_{i_n i_{n+1}} \leq m' - m - 1 \qquad \forall v \in V. \quad (5.69)$$

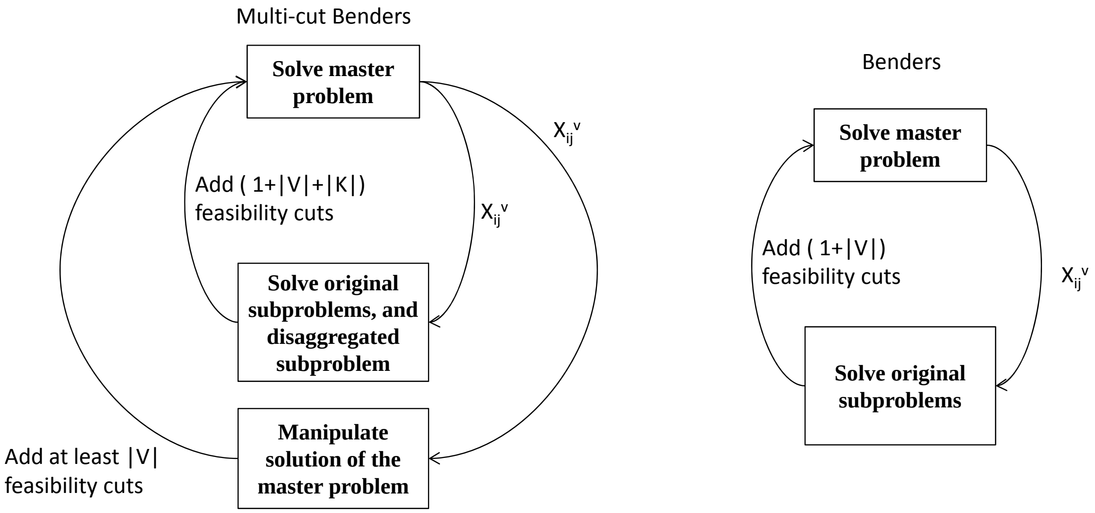
3. The delivery deadline of a commodity $k \in K$ is violated.

   In this case cut (5.70) is a strengthening of cut (5.66), where m denotes the placement of the destination of commodity k.

$$\sum_{n:i_n \in F, n \leq m-1} x^v_{i_n i_{n+1}} \leq m - 2 \qquad \forall v \in V. \quad (5.70)$$

This concludes the improvements made on the Benders decomposition algorithms. An overall illustration of the multi-cut Benders decomposition algorithm is given by Figure 5.9. Chapter 6 analyzes the continuous time approach to SNDP by several numerical tests.

Figure 5.9: Multi-cut Benders decomposition, and original Benders decomposition.


Multi-cut Benders
Solve master problem
Add (1+|V|+|K|) feasibility cuts
X_{ij}^v
X_{ij}^v
Solve original subproblems, and disaggregated subproblem
Manipulate solution of the master problem
Add at least |V| feasibility cuts
Benders
Solve master problem
Add (1+|V|) feasibility cuts
X_{ij}^v
Solve original subproblems

Chapter 6

Numerical Tests

The first part of the numerical tests is dedicated to the comparison of the periodic and continuous time approaches. In the first step, we compare the periodic approach with the continuous approach in terms of ease of solution under the same problem setting. We solve the periodic and continuous time models under the same original network setting, and analyze how each model performs in terms of solution time and optimality gap. In the next step we show how the quality of a solution decreases as we aggregate the problem into smaller numbers of periods.

The second part of the numerical tests focuses on analyzing the Benders decomposition approaches. We test solution time and quality of lower bounds when relaxing commodity constraints, time constraints, or time and commodity constraints. We then choose the most promising decomposition approach and apply all proposed improvements, and discuss their effects.

Tests instances are generated using the original network $G = (N, A)$. That is we consider an original network of terminals, with a determined planning horizon T, set of terminals N, a complete set of arcs A, set of commodities K, and set of vehicles V; and use it to build the networks for the periodic and continuous time approaches. Three problem sizes of $|K| = 5, 10, 15$ are considered with the number of vehicles set to $|V| = \lceil |K|/2 \rceil$. Furthermore we consider two network settings of congested and uncongested terminals. A congested terminal setting implies that a terminal $i \in N$ has a higher chance of serving as the origin or destination of more than one commodity $k \in K$ compared to an uncongested terminal. To generate a network with congested terminals we set $|N| = 5$ and randomly allocate all commodity origins and destinations to available terminals. For an uncongested network we set $|N| = 20$. After commodity allocation, we remove any node $i \in N$ that does not accommodate a commodity $k \in K$. For each data setting we generate five random instances. The cost, time, and demand are discussed separately in each section.

All models and algorithms are coded in Matlab R2011 and performed on a PC with Intel Xeon 3.00 GHz and 8G RAM. We impose a one hour CPU time limit, and use Gurobi 5.50 as the commercial solver.

6.1 Comparison of the periodic and continuous time models

This section compares the periodic and continuous time approaches to SNDP. The first section compares the ease of solution of the periodic and continuous time approaches. The periodic and continuous models are compared by solution time and quality of lower bound. The second section analyzes the impact of aggregating operations into discrete time periods on the quality of solutions obtained from [PM2].

In this section cost, time and demand settings are adopted from Bai et al. (2014).

Vehicle waiting cost is $c_{ii} = 100$, transportation cost c_{ij} is set as either 150 or 250. Each transportation move consumes one unit of time. Availability and delivery deadline times $\sigma(k) \leq \tau(k)$, are randomly generated in the planning horizon. Commodity demand d_k follows a triangular distribution of $(2, 14, 8)$, and vehicle capacity is $\eta = 20$.

6.1.1 Analyzing ease of solution

In this section we set the planning horizon to one week with hourly operations. In the periodic network each hour is interpreted as one period resulting in a total of $|T| = 168$ periods. In the continuous model each hour is interpreted as one unit of time. We compare models [CM2] and [PM2] in terms of network and model size, and solution time (Cpu) and quality. All comparisons are based on the average data of each size setting.

The network and model size is given in Tables 6.1 and 6.2. We compare the original network size (Original) and analyze its expansion in the time-space (Periodic time), and continuous time network. The continuous time network is presented with (Original continuous time) and without (Reduced continuous time) the network and model reductions discussed in Chapter 3. Note that the number of nodes is equal in both the original and reduced continuous time networks, and so, it has only be given in the original continuous time section.

The original network size is larger in the uncongested terminal setting, compared to the congested terminal setting. Averaging on all data sizes, in the uncongested terminal setting, the periodic network expands the number of nodes by 16,700%, while this expansion is 111% in the continuous time network. The number of arcs increases by 18,011% in the periodic time network, and by 139% in the reduced continuous time network. The periodic time model is 7,765% larger than the reduced continuous time model. The network and model reduction techniques reduce the number of arcs by 26%, and reduce model size by 39%.

Averaging on all data sizes, in the congested terminal setting, the periodic network expands the number of nodes by 16,700%, while the continuous time network increases the number of nodes by 396%. The number of arcs increases by 20,958% in the periodic time network, and by 1,549% in the reduced continuous time network. The periodic time model is 838% larger than the reduced continuous time model. The network and model reduction techniques reduce the number of arcs by 36%, and reduce model size by 40%.

The quality of a solution in the periodic or continuous time approach is determined by the lower bound (LB) and the upper bound (UB), reached by the commercial solver. The optimality gaps (Gap%) are calculated as $(UB-LB)/LB$, and are reported in percentages. An "∞" sign indicates that no value was derived from Gurobi within the 1 hour time limit. Results are given in Tables 6.3 and 6.4.

In uncongested terminal settings, the continuous model outperforms the periodic model in all aspects. In the smaller size instances of $|K| = 5$, the continuous model reaches the optimal solution within seconds. The periodic model has varying results and can result in gaps as high as 164.6% after one hour of computation. On average the continuous model gives 47.1% higher lower bounds than the periodic model. In other instance sizes of $|K| = 10, 15$, the periodic model does not reach any bound within the one hour CPU time limit. The continuous model reaches a lower bound in all instances and gives an upper bound in two of the ten instances. The lower bounds are, however, poor.

The periodic model performs better in congested terminals compared to uncongested terminals. This is due to the reduction in size of the network. However, the continuous time model still outperforms the periodic model in the small instance size of $|K| = 5$. The continuous model reaches better lower bounds by 63.1% while utilizing 99.9% lower CPU time. In the $|K| = 10$ instance size both models reach comparable results. The periodic model gives an average 17.7% lower optimality gaps and 9.7% better lower bounds while utilizing 19.2% higher CPU time compared to the continuous model. In the instance size of $|K| = 15$ both models perform poorly. The periodic model does not reach a bound in two of the five instances and gives poor lower bounds in the other three instances. The continuous model consistently reaches lower bounds, which are very poor.

Overall results show that both models are very hard to solve. The continuous time ease of solution is greatly affected by the number of commodities. This is due to the increase in network size. The periodic model cannot solve any problem under the number of assumed periods, even in small size instances of $|K| = 5$. For an efficient periodic model we can aggregate the operations into smaller numbers of periods. However, we show in the next section that this aggregation greatly affects solution quality, and cannot be used to model the same problem.

6.1.2 Analyzing quality of solution

In this section we analyze the periodic time model in terms of modeling a problem under different number of periods. The periodic model improves its performance when the number of periods $|T|$ decreases. Therefore, to have an efficient solution of real life problems, the planning horizon is divided into low numbers of periods. All services are then aggregated to the assumed periods. We show how this aggregation changes the quality of the solution.

We employ the cost, time, and demand settings of Bai et al. (2014). To have a better analysis and reach optimal solutions, we consider a congested network of terminals $|N| = 5$, a low number of periods $|T| = 20$, and $|K| = 10$; and aggregate the problem into the two cases $|T| = 10, 5$ periods. Aggregation is done by changing all availability times $\sigma(k)$, and delivery deadlines $\tau(k)$ to correspond to their new periods, and updating costs accordingly. As an example when aggregating from $|T| = 20$ to $|T| = 10$, availability times are updated as $\lceil \sigma(k)/2 \rceil$, similarly delivery deadlines are updated as $\lceil \tau(k)/2 \rceil$. Vehicle waiting time in the aggregated network $|T| = 10$ corresponds to double the vehicle waiting times in the disaggregated network $|T| = 20$; and so waiting cost is set to $2c_{ii}$. Transportation in the aggregated network $|T| = 10$ corresponds to a transportation move and one vehicle waiting time in the disaggregated network $|T| = 20$, and so transportation costs are set to $c_{ij} + c_{ii}$. Figures 6.1 and 6.2, illustrate the aggregation of the first instance for $|T| = 5$, $|T| = 10$, respectively. Figure 6.3 shows the optimal solution of $|T| = 20$ for the first instance.

Table 6.5 gives the results of aggregating $|T| = 20$ into $|T| = 10, 5$. Reducing the number of periods by 50% decreases solution time by an average 93.3% and increased the solution cost by an average 42.3%. Service is changed by using two vehicles compared to the optimal one vehicle. Reducing the number of periods by 75%, decreases solution time 99.9%, and increases solution cost by an average 62.2%. Service is changed by using three vehicles compared to the optimal one vehicle. In our opinion, the decrease in solution time does not compensate for the increase of solution cost.

6.2 Analysis of Benders decomposition

In this section we analyze the performance of the Benders decomposition approaches and test the effect of improvements on the most promising approach. The most promising approach is that which solves faster and gives better lower bounds. The major issue in our decomposition approaches is the solution time of the master problem. In that regard the decomposition approach with the minimum solution time of the master problem has the most potential for improvement. If the master problem takes a long time to solve without any improvement cuts, then their addition incurs a higher burden and does not aid in the decomposition process.

To investigate a broader set of instances, we generate a new set of values for cost, time and demand. We consider a one week planning horizon with hourly operations, $|T| = 168$. To determine transportation times, t_{ij}, we generate random points on a $|T|/3 \times |T|/3$ plane. Transportation times are equal to the euclidean distance between the generated points. The $T/3$ plane is selected to avoid transportation times greater than $\sqrt{2}T/3$ (near the half duration of the planning horizon), as they are unrealistic. We consider two settings for the window $[\sigma(k), \tau(k)]$ of a commodity $k \in K$. For a loose time window, we set $\tau(k) = \sigma(k) + 4t_{ij}$, and for a tight time window we set $\tau(k) = \sigma(k) + 2t_{ij}$. The availability times $\sigma(k)$ are randomly chosen in the interval $[0, T - 4t_{ij}]$ for loose windows and $[0, T - 2t_{ij}]$ for tight windows. Vehicle waiting cost is $c = 1$ per unit of time, and transportation costs are set to $c_{ij} = 10t_{ij}$. Demand d_k is uniformly generated in interval $[1, 5]$. Vehicle capacity is set to $\eta = 20$.

6.2.1 Comparison of Benders decomposition approaches

We analyze the results of the classical Benders decomposition when relaxing commodity constraints (3.25)-(3.28), relaxing time constraints (3.29)-(3.32), and relaxing time and commodity constraints (3.3)-(3.14). We seek to choose the decomposition approach with the most potential to implement the improvements of Chapter 5. We run each approach on the instances with loose time windows and uncongested terminals as initial experiments showed they are the hardest instances to solve. We also only present the results of the largest instances with $|K| = 15$ as the differences of the decomposition approaches are more apparent.

The results of each decomposition approach are given in Table 6.6. Decomposing time constraints doesn't seem to be an efficient solution procedure. Although the lower bounds given by the master problem are the strongest between all approaches, the master problem is very large and utilizes a high CPU time to solve. In fact the relaxed master problem [RMPt] is not solved to optimality even for one iteration. Our aim in using Benders decomposition is to make the problem easier to solve, and adding more constraints to [RMPt] as improvements only increase the difficulty in solving a still very hard problem. Relaxing commodity constraints relatively decreases the solution burden of the master problem [RMPy]. When relaxing both time and commodity constraints, the relaxed master problem [RMPyt] is relatively easy to solve. The average lower bound is greater than the average lower bound when relaxing commodity constraints. We believe that relaxing time and commodity constraints has the most potential for improvement as it may accommodate the improvement cuts better than the other two approaches.

6.2.2 Effects of algorithm improvements

In the final section of the numerical test we analyze the effects of improvements in the Benders algorithm when relaxing both time and commodity constraints (3.25)-(3.32). We test on four data sets with different settings in terminal congestion and windows $[\sigma(k), \tau(k)]$. We compare the solution of the commercial solver to the classical Benders approach with no improvement cuts implemented, Benders decomposition with all improvement cuts except the modified MTZ cuts (5.22)-(5.28), and Benders decomposition with all improvement cuts including the modified MTZ subtour elimination and precedence cuts (5.22)-(5.28). The modified MTZ subtour elimination and precedence cuts are the strongest of the cuts introduced, and we seek to observe the trade-off between ease of solution and solution quality when they are introduced into the master problem. When including MTZ cuts the solution of the master problem is always feasible in subproblems [SPvt]. This relaxation is equivalent to replacing time constraints (3.29)-(3.32) by the MTZ constraints (5.22)-(5.28), and relaxing commodity constraints (3.25)-(3.28). Tables 6.7-6.10 give the results of the comparison.

In small sized instances of $|K| = 5$, all approaches, except classical Benders decomposition, consistently reach the optimal solution. Improved Benders decomposition with all improvement cuts reaches the optimal solution in an average 1.2 iterations and reduces the required CPU time to an average 2 seconds which is a 99.2% improvement to the solver. The effects of including modified MTZ cuts are most apparent in $|K| = 5$. Results show that including these cuts decreases the number of iterations and CPU time by an average 94.3% and 98.1% respectively.

In instances of size $|K| = 10$, the solver and classical Benders decomposition do not solve any instances in the one hour time limit. Benders decomposition with all improvement cuts reaches the optimal solution in 10 of the 20 instances in an average 2,125 seconds. The hardest instances are uncongested terminals with loose windows, where no instances are solved. When removing the modified MTZ cuts the master problem becomes easier to solve as indicated by the increase in number of iterations within the one hour time limit. The increase in speed does not, however, compensate for quality of solutions. Five instances are solved compared to the previous 10, and the lower bound quality is decreased by 3.7%.

In the largest instance size of $|K| = 15$, no approach is able to solve any of the instances. The classical Benders approach remains the weakest approach and gives the lowest average lower bound. Bender decomposition with all improvements and no modified MTZ cuts is the strongest, with an average increase of 68.1% and 4.9% in the lower bounds compared to the solver and Benders with all improvements, respectively. Adding the modified MTZ cuts to the master problem increases its solution difficulty. The master problem is not solved to optimality even for one iteration when including the modified MTZ cuts, and the lower bound cannot be checked for overall optimality in the subproblems.

This concludes the numerical tests of Chapter 6. In summary the numerical results show that the continuous time approach may lead to a better and more effective modeling of SNDP compared to the periodic time approach. In terms of solution approach, Benders decomposition when reformulating both time and commodity constraints leads to faster solutions of the relaxed master problem, and can be considerably strengthened by the proposed algorithm improvements. Chapter 7 concludes the thesis.

Table 6.1: Network and model size - Uncongested terminals

		Original		Periodic Time			Original continuous Time			Reduced continuous Time				
	Inst	$\|N\|$	$\|A\|$	$\|N^p\|$	$\|A^p\|$	#Var	$\|N^c\|$	$\|A^c\|$	#Var	$\|A^c\|$	$\|A^c_1\|$	$\|A^c_2\|$	$\|A^c_3\|$	#Var
$\| K \|= 5$	1	10	90	1,680	16,800	302,400	15	120	2,160	94	64	0	30	1,524
	2	8	56	1,344	10,752	193,536	15	120	2,160	65	34	1	30	909
	3	9	72	1,512	13,608	244,944	15	120	2,160	107	77	0	30	1,797
	4	8	56	1,344	10,752	193,536	15	120	2,160	92	60	2	30	1,470
	5	9	72	1,512	13,608	244,944	14	115	2,070	61	36	0	25	906
	Avr	**8.8**	**69.2**	**1,478.4**	**13,104.0**	**235,872.0**	**14.8**	**119.0**	**2,142.0**	**83.8**	**54.2**	**0.6**	**29.0**	**1,321.2**
$\| K \|= 10$	1	15	210	2,520	37,800	2,079,000	27	460	25,300	312	229	3	80	14,960
	2	15	210	2,520	37,800	2,079,000	27	460	25,300	314	231	3	80	14,810
	3	12	132	2,016	24,192	1,330,560	26	450	24,750	363	288	5	70	18,230
	4	15	210	2,520	37,800	2,079,000	30	490	26,950	386	275	1	110	17,650
	5	14	182	2,352	32,928	1,811,040	28	470	25,850	419	325	4	90	20,600
	Avr	**14.2**	**188.8**	**2,385.6**	**34,104.0**	**1,875,720.0**	**27.6**	**466.0**	**25,630.0**	**358.8**	**269.6**	**3.2**	**86.0**	**17,250.0**
$\| K \|= 15$	1	14	182	2,352	32,928	4,214,784	40	1,035	132,480	742	571	6	165	81,016
	2	17	272	2,856	48,552	6,214,656	39	1,020	130,560	634	480	4	150	68,160
	3	17	272	2,856	48,552	6,214,656	40	1,035	132,480	797	626	6	165	88,496
	4	17	272	2,856	48,552	6,214,656	41	1,050	134,400	722	534	8	180	76,464
	5	15	210	2,520	37,800	4,838,400	40	1,035	132,480	730	559	6	165	79,384
	Avr	**16.0**	**241.6**	**2,688.0**	**43,276.8**	**5,539,430.4**	**40.0**	**1.035.0**	**132,480.0**	**725.0**	**554.0**	**6.0**	**165.0**	**78,704.0**

Table 6.2: Network and model size - Congested terminals

		Original		Periodic Time			Original continuous Time			Reduced continuous Time				
	Inst	$\|N\|$	$\|A\|$	$\|N^p\|$	$\|A^p\|$	#Var	$\|N^c\|$	$\|A^c\|$	#Var	$\|A^c\|$	$\|A_1^c\|$	$\|A_2^c\|$	$\|A_3^c\|$	#Var
$\| K \|= 5$	1	4	12	672	2,688	48,384	14	115	2,070	76	49	2	25	1,209
	2	5	20	840	4,200	75,600	14	115	2,070	77	51	1	25	1,236
	3	5	20	840	4,200	75,600	15	120	2,160	64	34	0	30	894
	4	5	20	840	4,200	75,600	13	110	1,980	77	54	3	20	1,299
	5	5	20	840	4,200	75,600	14	115	2,070	66	40	1	25	1,005
	Avr	**4.8**	**18.4**	**806.4**	**3,897.6**	**70,156.8**	**14.0**	**115.0**	**2,071.8**	**72.0**	**45.6**	**1.4**	**25.0**	**1,128.6**
$\| K \|= 10$	1	5	20	840	4,200	231,000	24	430	23,650	307	242	15	50	15,770
	2	5	20	840	4,200	231,000	24	430	23,650	184	120	14	50	8,400
	3	5	20	840	4,200	231,000	25	440	24,200	291	220	11	60	14,350
	4	5	20	840	4,200	231,000	25	440	24,200	196	130	6	60	8,700
	5	5	20	840	4,200	231,000	25	440	24,200	264	193	11	60	12,730
	Avr	**5.0**	**20.0**	**840.0**	**4,200.0**	**231,000.0**	**24.6**	**436.0**	**23,980.0**	**248.4**	**181.0**	**11.4**	**56.0**	**11,990.0**
$\| K \|= 15$	1	5	20	840	4,200	537,600	35	960	122,880	594	470	34	90	69,440
	2	5	20	840	4,200	537,600	35	960	122,880	666	546	30	90	79,296
	3	5	20	840	4,200	537,600	35	960	122,880	590	471	29	90	68,976
	4	5	20	840	4,200	537,600	34	945	120,960	646	512	59	75	77,912
	5	5	20	840	4,200	537,600	35	960	122,880	717	586	41	90	86,056
	Avr	**5.0**	**20.0**	**840.0**	**4,200.0**	**537,600.0**	**34.8**	**957.0**	**122,496.0**	**642.6**	**517.0**	**38.6**	**87.0**	**76,336.0**

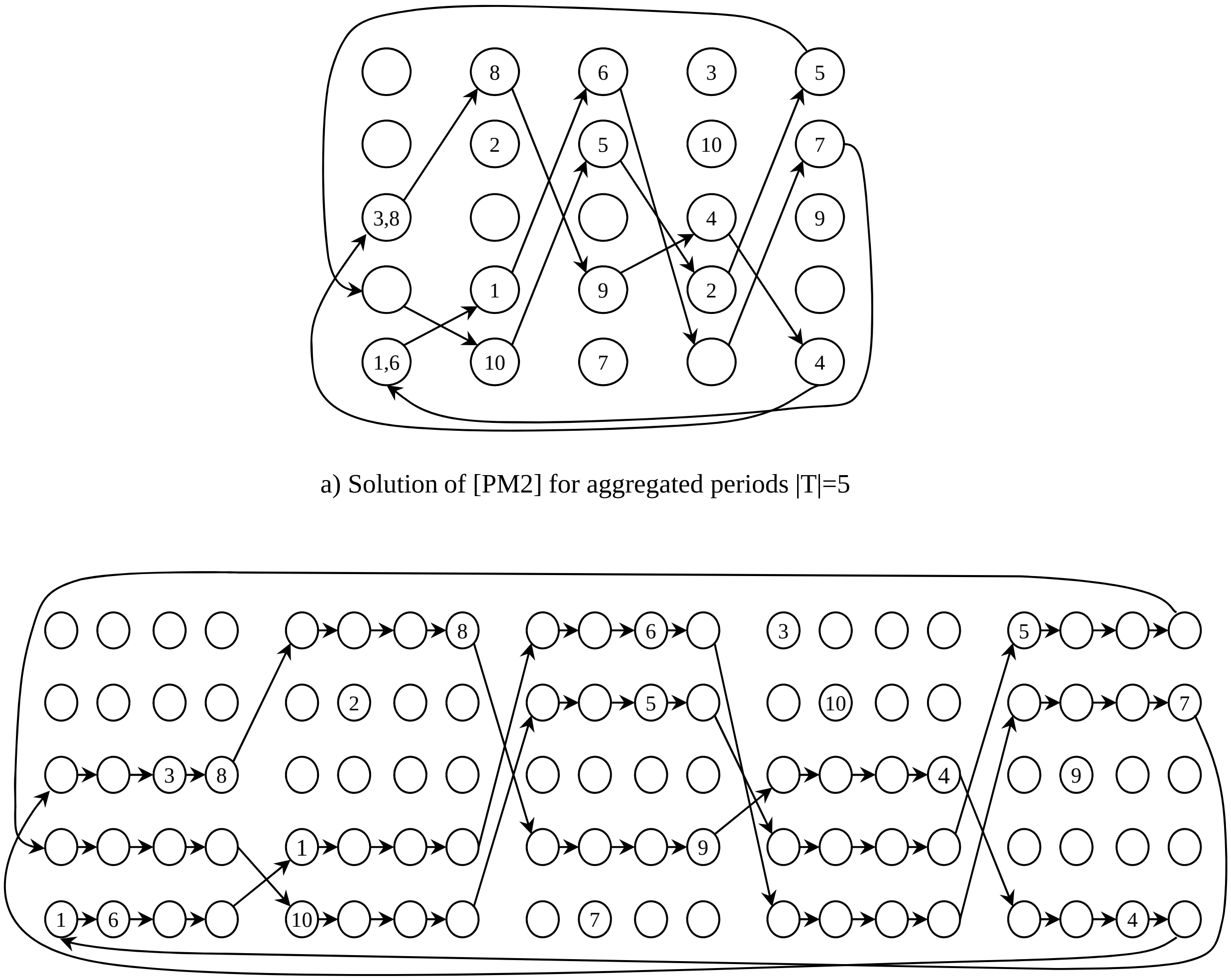
Table 6.3: Comparison of [PM2] and [CM2] - Uncongested terminals

		Periodic Model				Continuous Model			
	Inst	Cpu	Gap%	LB	UB	Cpu	Gap%	LB	UB
$\mid K \mid= 5$	1	3622	164.6	6,650	17,600	4	0	17,200	17,200
	2	4569	95.5	8,950	17,500	0	0	17,250	17,250
	3	2989	1.2	17,100	17,300	75	0	17,250	17,250
	4	4907	0.9	17,200	17,350	17	0	17,350	17,350
	5	3388	96.1	8,850	17,350	3	0	17,350	17,350
	Avr	**3895**	**71.7**	**11,750**	**17,420**	**20**	**0**	**17,280**	**17,280**
$\mid K \mid= 10$	1	3678	∞	-∞	∞	3416	518.9	2,900	17,950
	2	3662	∞	-∞	∞	3349	∞	2,750	∞
	3	3661	∞	-∞	∞	3411	∞	2,300	∞
	4	3621	∞	-∞	∞	3314	1,280.4	2,550	35,200
	5	3679	∞	-∞	∞	3355	∞	2,200	∞
	Avr	**3660**	**-**	**-**	**-**	**3369**	**899.7**	**2,540**	**26,575**
$\mid K \mid= 15$	1	3812	∞	-∞	∞	3493	∞	17,850	∞
	2	3816	∞	-∞	∞	3429	∞	3,600	∞
	3	3809	∞	-∞	∞	3492	∞	1,900	∞
	4	3805	∞	-∞	∞	3495	∞	3,400	∞
	5	3811	∞	-∞	∞	3493	∞	3,150	∞
	Avr	**3811**	**-**	**-**	**-**	**3480**	**-**	**5,980**	**-**

Table 6.4: Comparison of [PM2] and [CM2] - Congested terminals

		Periodic Model				Continuous Model			
	Inst	Cpu	Gap%	LB	UB	Cpu	Gap%	LB	UB
$\mid K \mid= 5$	1	3444	94.9	8,900	17,350	9	0	17,300	17,300
	2	6459	91.2	9,100	17,400	2	0	17,300	17,300
	3	6456	0.9	17,050	17,200	1	0	17,200	17,200
	4	5792	97.1	8,750	17,250	6	0	17,150	17,150
	5	3589	94.5	9,050	17,600	0	0	17,250	17,250
	Avr	**5148**	**75.7**	**10,570**	**17,360**	**3.6**	**0**	**17,240**	**17,240**
$\mid K \mid= 10$	1	3388	905.4	1,850	18,600	3391	1,048.4	1,550	17,800
	2	3386	5.4	17,450	18,400	3521	96	17,600	34,500
	3	3387	205.0	5,950	18,150	3430	963.6	1,650	17,550
	4	3492	7.0	17,700	18,950	498	0	18,200	18,200
	5	3387	1,427.0	1,650	25,200	3444	990.9	1,650	18,000
	Avr	**3408**	**510.0**	**8,920**	**19,860**	**2857**	**619.8**	**8,130**	**21,210**
$\mid K \mid= 15$	1	3371	184.5	6,450	18,350	3093	3,387.5	2,000	69,750
	2	3718	∞	-∞	∞	3095	∞	2,000	∞
	3	3452	∞	9,000	∞	3493	∞	2,200	∞
	4	3376	∞	7,500	∞	3491	2,270.5	2,200	52,150
	5	3681	∞	-∞	∞	3492	∞	1,800	∞
	Avr	**3520**	**184.5**	**7,650**	**18,350**	**3333**	**2,829.0**	**2,000**	**60,950**

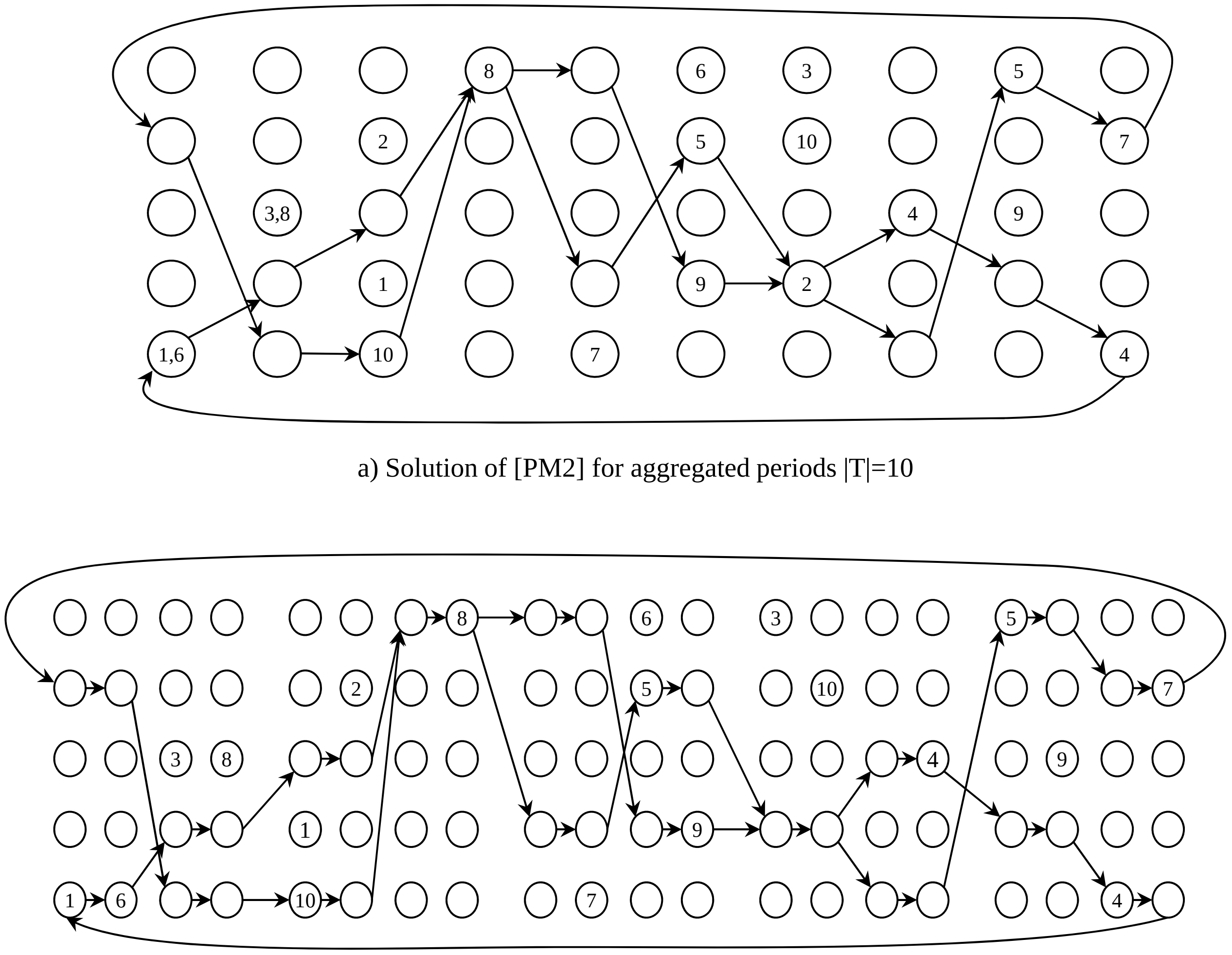
Figure 6.1: Aggregation of periods from $|T| = 20$ to $|T| = 5$.



a) Solution of [PM2] for aggregated periods |T|=5

b) Corresponding disaggregated solution for periods |T|=20

Figure 6.2: Aggregation of periods from $|T| = 20$ to $|T| = 10$.



a) Solution of [PM2] for aggregated periods |T|=10

b) Corresponding disaggregated solution for periods |T|=20

Figure 6.3: Optimal solution of [PM2] for disaggregated network of $|T| = 20$.

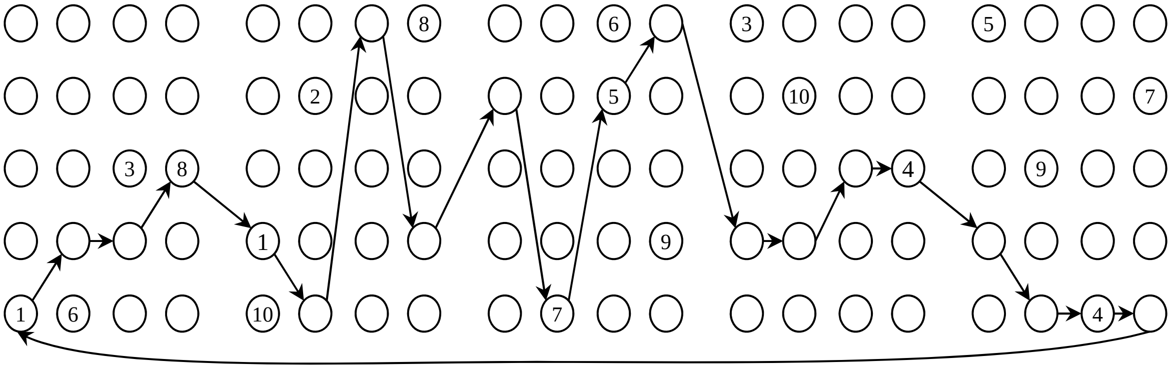


Table 6.5: Quality of [PM2] solutions with varying aggregation levels.

	$\mid T \mid= 20$			$\mid T \mid= 10$			$\mid T \mid= 5$		
Inst	Cpu	Gap%	Cost	Cpu	Gap%	Cost	Cpu	Gap%	Cost
1	5687	0	2,900	267	0	4,950	3	0	7,450
2	1906	0	2,750	206	0	4,900	3	0	6,700
3	8327	0	2,750	818	0	5,050	4	0	7,300
4	4114	0	2,950	209	0	4,900	2	0	9,050
5	3100	0	2,900	45	0	4,900	4	0	7,200
Avr	**4627**	**0**	**2,850**	**309**	**0**	**4,940**	**3**	**0**	**7,540**

Table 6.6: Comparison of Benders decomposition approaches

	Relaxing commodity constraints				Relaxing time constraints				Relaxing time and commodity constraints			
Inst	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB
1	10	4503	-	768	1	4981	-	1,488	23	3769	-	744
2	8	6399	-	678	1	9638	-	1,386	22	4819	-	582
3	8	4595	-	861	1	7193	-	1,575	30	4092	-	1,005
4	6	3687	-	768	1	5682	-	1,197	28	3619	-	888
5	14	4070	-	708	1	3771	-	1,734	37	3633	-	915
Avr	**9.2**	**4650**	**-**	**756.6**	**1**	**6253**	**-**	**1,476**	**28**	**3986**	**-**	**826.8**

Table 6.7: Effect of improvements - Congested terminals - Tight windows

		Solver				Classic Benders				Improved Benders				Improved Benders + MTZ cuts			
	Inst	Cpu	Gap%	LB	UB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB
$\mid K \mid= 5$	1	157	0	1,929	1,929	148	2930	0	1,929	5	6	0	1,929	2	1	0	1,929
	2	180	0	2,853	2,853	157	3608	-	2,700	9	11	0	2,853	1	1	0	2,853
	3	5	0	1,947	1,947	17	5	0	1,947	3	2	0	1,947	1	1	0	1,947
	4	10	0	1,326	1,326	56	83	0	1,326	9	2	0	1,326	1	1	0	1,326
	5	58	0	1,758	1,758	118	979	0	1,785	15	12	0	1,758	2	1	0	1,758
	Avr	**82**	**0**	**1,962.6**	**1962.6**	**99.2**	**1521**	**0**	**1,932**	**8.2**	**7**	**0**	**1,962.6**	**1.2**	**1**	**0**	**1,926.0**
$\mid K \mid= 10$	1	3840	∞	1,956	∞	150	3693	-	1,947	1	5	0	2,592	2	7	0	2,592
	2	3880	∞	2,280	∞	94	3667	-	2,127	37	1764	0	2,898	2	87	0	2,898
	3	3878	∞	1,920	∞	131	3647	-	2,118	19	3981	-	4,203	4	1735	0	4,965
	4	3891	∞	1,077	∞	256	3602	-	750	9	5417	-	1,932	1	3946	-	1,668
	5	3915	∞	1,752	∞	141	3684	-	1,848	9	66	0	2,601	4	149	0	2,601
	Avr	**3681**	**-**	**1,797**	**-**	**154.4**	**3659**	**-**	**1,754.4**	**15.0**	**2,247**	**0**	**2845.2**	**2.6**	**1185**	**0**	**2,944.8**
$\mid K \mid= 15$	1	3492	∞	1,548	∞	81	3607	-	1,176	4	3621	-	3,165	1	3955	-	2,952
	2	3492	∞	1,461	∞	44	3714	-	1,194	4	6220	-	3,132	1	3957	-	2,829
	3	3492	∞	1,905	∞	73	3750	-	1,311	6	6453	-	2,301	1	3652	-	3,030
	4	3698	∞	1,476	∞	83	3609	-	1,338	2	3313	-	2,145	1	3954	-	2,256
	5	3493	∞	1,146	∞	54	3640	-	1,272	4	3940	-	2,142	1	3935	-	1,884
	Avr	**3533**	**-**	**1,507.2**	**-**	**67**	**3664**	**-**	**1,258.2**	**4.0**	**4709**	**-**	**2,577**	**1**	**3891**	**-**	**2,590.2**

Table 6.8: Effect of improvements - Uncongested terminals - Tight windows

		Solver				Classic Benders				Improved Benders				Improved Benders + MTZ cuts			
	Inst	Cpu	Gap%	LB	UB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB
$\mid K \mid= 5$	1	24	0	2,100	2,100	187	3606	-	1,938	20	28	0	2,100	1	1	0	2,100
	2	58	0	3,231	3,231	130	1682	0	3,231	18	23	0	3,231	1	1	0	3,231
	3	612	0	2,649	2,649	192	3624	-	2,088	14	17	0	2,649	1	0	0	2,649
	4	3	0	2,349	2,349	32	17	0	2,349	5	1	0	2,349	3	0	0	2,349
	5	2963	0	2,775	2,775	250	3612	-	2,034	5	6	0	2,775	1	2	0	2,775
	Avr	**732**	**0**	**2,620.8**	**2,620.8**	**158.2**	**2508**	**0**	**2,328.0**	**12.4**	**15**	**0**	**2,620.8**	**1.4**	**1**	**0**	**2,620.8**
$\mid K \mid= 10$	1	3891	∞	2,118	∞	70	3691	-	1,737	7	303	0	3,582	1	48	0	3,528
	2	3824	∞	1,752	∞	69	3721	-	1,956	8	4292	-	3,894	1	3934	-	3,630
	3	3695	∞	1,671	∞	77	3642	-	1,212	17	3858	-	3,237	1	856	0	3,678
	4	3820	∞	2,115	∞	63	3643	-	1,893	17	3956	-	3,306	1	598	0	3,675
	5	3689	∞	1,989	∞	83	3713	-	1,785	13	4075	-	3,615	1	3941	-	3,804
	Avr	**3784**	**-**	**1,929**	**-**	**72.4**	**3682**	**-**	**1,711.2**	**12.4**	**3297**	**0**	**3,526.8**	**1**	**1876**	**0**	**3,663.0**
$\mid K \mid= 15$	1	3491	∞	2,019	∞	66	3658	-	903	1	3983	-	3,861	1	3943	-	3,504
	2	3492	∞	1,872	∞	38	3250	-	1,722	3	4273	-	4,149	1	3935	-	3,495
	3	3492	∞	2,151	∞	46	3758	-	1,587	6	4761	-	3,633	1	3924	-	3,300
	4	3492	∞	1,635	∞	41	3724	-	1,266	3	3887	-	4,002	1	3887	-	3,600
	5	3493	∞	1,677	∞	56	3769	-	1,554	6	4510	-	3,561	1	4456	-	3,510
	Avr	**3492**	**-**	**1,870.8**	**-**	**49.4**	**3632**	**-**	**1,406.4**	**3.8**	**4283**	**-**	**3,841.2**	**1**	**4029**	**-**	**3,481.8**

Table 6.9: Effect of improvements - Congested terminals - Loose windows

		Solver				Classic Benders				Improved Benders				Improved Benders + MTZ cuts			
	Inst	Cpu	Gap%	LB	UB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB
$\mid K \mid= 5$	1	4	0	1,077	1,077	26	3	0	1,077	9	2	0	1,077	1	1	0	1,077
	2	6	0	1,164	1,164	40	17	0	1,164	12	3	0	1,164	1	1	0	1,164
	3	1	0	1,104	1,104	41	8	0	1,104	4	1	0	1,104	2	1	0	1,104
	4	11	0	1,131	1,131	62	45	0	1,131	22	37	0	1,131	1	2	0	1,131
	5	7	0	1,239	1,239	28	5	0	1,239	5	1	0	1,239	1	1	0	1,239
	Avr	**6**	**0**	**1,143**	**1,143**	**39.4**	**16**	**0**	**1,143**	**10.4**	**9**	**0**	**1,143.0**	**1.2**	**1.0**	**0**	**1,143.0**
$\mid K \mid= 10$	1	3839	89.8	1,035	1,965	189	3629	-	1,176	22	1112	0	1,230	1	62	0	1,230
	2	3647	∞	864	∞	96	3693	-	969	24	4715	-	1,113	2	2256	-	1,353
	3	3718	∞	684	∞	204	3628	-	582	88	3622	-	1,005	1	3945	-	996
	4	3798	∞	1,089	∞	171	3680	-	1,005	112	3656	-	1,320	1	921	0	1,560
	5	3813	32.9	903	1,200	157	3601	-	1,023	210	3611	-	1,059	2	345	0	1,200
	Avr	**3763**	**61.4**	**907.8**	**1,582.5**	**163.4**	**3646**	**-**	**951**	**91.2**	**3343**	**-**	**1,145.4**	**1.4**	**1506**	**0**	**1,267.8**
$\mid K \mid= 15$	1	3592	∞	852	∞	28	3694	-	834	20	3618	-	1,311	1	3938	-	1,176
	2	3593	∞	996	∞	80	3606	-	1,032	12	4474	-	1,068	1	3920	-	1,152
	3	4332	∞	1,299	∞	65	3639	-	1,050	50	3716	-	1,410	1	3923	-	1,410
	4	3594	251.0	1,035	3,633	62	3701	-	726	35	3684	-	1,194	1	3912	-	1,032
	5	3591	∞	1,032	∞	59	3647	-	1,050	28	4787	-	1,212	1	3933	-	1,137
	Avr	**3740**	**251.0**	**1,042.8**	**3,633**	**58.8**	**3657**	**-**	**938.4**	**29**	**4056**	**-**	**1,212**	**1**	**3925**	**-**	**1,181.4**

Table 6.10: Effect of improvements - Uncongested terminals - Loose windows

				Solver			Classic Benders				Improved Benders				Improved Benders + MTZ cuts		
	Inst	Cpu	Gap%	LB	UB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB	Iter	Cpu	Gap%	LB
$\mid K \mid = 5$	1	327	0	1,965	1,965	169	3667	-	1,929	80	1146	0	1,965	1	10	0	1,965
	2	52	0	1,848	1,848	116	589	0	1,848	52	148	0	1,848	1	7	0	1,848
	3	169	0	1,740	1,740	108	1169	0	1,740	56	269	0	1,740	1	4	0	1,740
	4	19	0	1,122	1,122	44	35	-	1,122	5	2	0	1,122	1	1	0	1,122
	5	184	0	1,437	1,437	144	3290	0	1,437	77	407	0	1,437	1	4	0	1,437
	Avr	**150**	**0**	**1,622.4**	**1,622.4**	**116.2**	**1750**	**0**	**1,615.2**	**54.0**	**394**	**0**	**1,622.4**	**1**	**5.2**	**0**	**1,622.4**
$\mid K \mid = 10$	1	3624	∞	1,470	∞	64	3735	-	1,365	55	3627	-	1,833	1	3918	-	1,929
	2	3812	∞	1,461	∞	65	3642	-	1,494	41	3670	-	1,884	1	3937	-	1,854
	3	3735	∞	1,761	∞	91	3712	-	1,275	23	4495	-	2,268	1	3951	-	2,265
	4	3793	∞	1,274	∞	79	3629	-	1,131	46	3868	-	1,635	1	3925	-	1,686
	5	3825	∞	1,518	∞	63	3657	-	1,212	24	3843	-	2,253	1	3930	-	2,184
	Avr	**3758**	**-**	**1,496.8**	**-**	**72.4**	**3675**	**-**	**1,295.4**	**37.8**	**3899**	**-**	**1,974.6**	**1**	**3932**	**-**	**1,983.6**
$\mid K \mid = 15$	1	3491	∞	684	∞	23	3769	-	744	13	4185	-	1,815	1	3987	-	1,761
	2	3492	∞	1,287	∞	22	4819	-	582	8	3694	-	1,743	1	3895	-	1,779
	3	3495	∞	1,257	∞	30	4092	-	1,005	16	4102	-	1,866	1	3872	-	1,785
	4	3593	∞	1,314	∞	28	3619	-	888	14	3948	-	1,644	1	3922	-	1,557
	5	3593	∞	1,689	∞	37	3633	-	915	8	3695	-	2,397	1	3988	-	2,232
	Avr	**3533**	**-**	**1,246.2**	**-**	**28.0**	**3986**	**-**	**826.8**	**11.8**	**3925**	**-**	**1895.0**	**1**	**3932**	**-**	**1,822.8**

Chapter 7

Conclusion

This thesis studies a continuous time approach to SNDP. A continuous time approach generalizes the periodic time approach in the literature by modeling the problem without the assumption of discrete time periods. To define the continuous time model for SNDP, we first develop a continuous time network which is reduced by several network reduction techniques. We then develop a continuous time model and reduce it using the characterisitcs of SNDP. Benders decomposition is used to decompose the continuous model in three directions. We may relax commodity constraints, relax time constrains, or relax both time and commodity constraints. To improve the classical Benders decomposition on SNDP we develop a number of algorithm improvements. We reduce the model based on the characterisitcs of SNDP, develop families of cuts to improve the lower bound of the master problems, and employ a multi-cut Benders algorithm.

We perform a number of numerical tests and show the advantages of the continuous time approach to SNDP. We compare the periodic and continuous time models in terms of ease of solution. The continuous time model outperforms the periodic model by as much as

99.9% in solution time when solving the same problem. It is shown that the Periodic model is unable to solve any of the considered instances to optimality. An efficient periodic model requires a low number of periods, and must aggregate real-life operations to the number of assumed periods. We show how aggregation can reduce solution quality of the periodic model both in terms of cost and service.

We test and analyze the ease of solution of the proposed Benders decomposition approaches. We show that relaxing both time and commodity constraints provides a better potential for improvement, as it is solved faster than the other two approaches, and provides relatively good lower bounds. The effects of improvements are then tested by comparing the solution of the continuous model by a commercial solver, to the classic Benders decomposition, Benders decomposition with all improvement cuts except subtour elimination and precedence cuts, and Benders decomposition with all improvement cuts. A classical Benders does not provide satisfactory lower bounds and is mostly outperformed by the commercial solver. The effect of improvements is apparent in the improved Benders decomposition, which outperforms the solver in all instances by substantial amounts.

There are a number of possible future research directions. The master problems of Benders decomposition approaches are very hard to solve. Therefore, any method that enables a faster solution to the master problem is an improvement. Dynamic programming approaches may be used to derive feasible vehicle cycles to the master problem and check optimality in the Benders subproblems. The master problem may be transformed into a path based formulation and solved by branch and price. Such an approach is used for the periodic time model studied in the literature. Employing a Benders branch and cut algorithm is another interesting approach. Finally, efficient heuristics may be developed for fast solutions to SNDP and to provide tight upper bounds.

References

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

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

Armacost, A. P., Barnhart, C., and Ware, K. A. Composite variable formulations for express shipment service network design. *Transportation science*, 36(1):1–20, 2002.

Ascheuer, N., Fischetti, M., and Grötschel, M. A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks*, 36(2):69–79, 2000.

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

Codato, G. and Fischetti, M. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.

Crainic, T. G. and Rousseau, J.-M. Multicommodity, multimode freight transportation:

A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3):225–242, 1986.

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

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

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

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

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

Kim, D., Barnhart, C., Ware, K., and Reinhardt, G. Multimodal express package delivery: A service network design application. *Transportation Science*, 33(4):391–407, 1999.

Lai, M. and Lo, H. K. Ferry service network design: optimal fleet size, routing, and scheduling. *Transportation Research Part A: Policy and Practice*, 38(4):305–328, 2004.

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

Meng, Q. and Wang, S. Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47 (5):695–708, 2011.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

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

Sung, C. and Song, S. Integrated service network design for a cross-docking supply chain network. *Journal of the Operational Research Society*, 54(12):1283–1295, 2003.

Teypaz, N., Schrenk, S., and Cung, V.-D. A decomposition scheme for large-scale service network design with asset management. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):156–170, 2010.

Wang, D. Z. and Lo, H. K. Multi-fleet ferry service network design with passenger preferences for differential services. *Transportation Research Part B: Methodological*, 42(9): 798–822, 2008.

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