Optimization of User Equilibrium container transportation problems using toll pricing

L.A.M. Bruijns

Graduation Thesis
MSc Applied Mathematics, TU Delft

Optimization of User Equilibrium container transportation problems using toll pricing

by

L.A.M. Bruijns

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday September 13, 2018 at 10:00 AM.

Student number:	4227905	
Project duration:	September 4, 2017 – September 13, 2018	
Thesis committee:	Prof. dr. ir. K. I. Aardal,	TU Delft, supervisor
	Dr. J. L. A. Dubbeldam,	TU Delft
	Dr. F. Phillipson,	TNO
	Ir. A. Sangers,	TNO

Abstract

The goal of this thesis is to provide a method to obtain a User Equilibrium (UE) in a synchromodal transportation network in which we transport containers for multiple customers.
We use the Space Time Network (STN), in which the travel time of modalities is fixed and independent of the occupancy of the network. We define the User Equilibrium solution as the solution wherein each customer can travel via their cheapest paths possible, and no customer is harmed by the route choice of other customers. We define the System Optimal (SO) solution as the solution in which the total costs of the network are minimized.
We expand the goal of finding a UE on the STN to finding a solution where the solution is both SO as well as UE. The solving method we developed in this thesis, consists of finding SO solutions by solving the Minimum Cost Multi-Commodity Flow (MCMCF) problem on the STN to find an SO solution, and create tolls schemes to create a UE solution, while maintaining the SO solution. We investigate several types of tolls in order to obtain sufficient both SO and UE solutions.

Preface

This thesis has been submitted as the final requirement to obtain the degree Master of Science in Applied Mathematics at Delft University of Technology. The research was conducted in collaboration with TNO in the period from September 2017 until September 2018.

During the last year, it was a real pleasure to work on my thesis at the Cyber Security & Robustness department of TNO with intelligent and nice people. First of all, I would like to thank Alex Sangers, Frank Phillipson and Karen Aardal for providing me the chance to work at my thesis at TNO and for your supervision during my internship. I would also like to thank Johan Dubbeldam for his willingness to be part of my thesis committee.

I would also like to express my gratitude to Irina and Kishan, my fellow interns in the Synchromodality project, and all of the other interns during my period at TNO. Thanks to all for making our time together a time to never forget.

Finishing this thesis feels like the end of an era. Six years ago I started my study Mathematics in Delft, a new study in a new city. I would like to thank my dear family for letting me make this decision, which brought me nothing but good things, and for their love and support during my whole life. Lastly I want to thank all my friends from both Krommenie and Delft, my student years not have been the same without them.

L.A.M. Bruijns
Delft, September 2018

Contents

List of used notation

$\mathcal{G} = (\mathcal{V}, \mathcal{A})$	directed graph
$v \in \mathcal{V}$	nodes
$a \in \mathcal{A}$	directed arcs
$a = (i, j)$	directed arc from i to j
x_a	arc flow
OD-pair	Origin-Destination-pair
$w \in \mathcal{W}$	OD-pairs / orders
$w = (w_O, w_D)$	OD-pair
d_w	demand for order w
$p \in \mathcal{P}_w$	paths for OD-pair w
$\mathcal{P} := \bigcup_{w \in \mathcal{W}} \mathcal{P}_w$	set of all paths
f_p	path flow
$f_{(w_O, w_D)}$	path flow on shortest path between w_O and w_D
δ_{ap}	incidence matrix entry
c_a	arc cost
m_a	arc capacity
$C^p,\ C_w^p$	path cost
$C_w = \sum_{p \in \mathcal{P}_w} C^p f_p = \sum_{p \in \mathcal{P}_w} C_w^p f_p$	order cost
m_p	path capacity
t	time step
l	location
s_w	start point for order w
e_w	end point for order w
β^p, β_w^p	toll per path
$\beta_w = \sum_{p \in \mathcal{P}_w} \beta^p f_p = \sum_{p \in \mathcal{P}_w} \beta_w^p f_p$	toll per order
$C_{\beta w}^p$	path cost including toll
$C_{\beta w}$	order cost including toll
k_w	cost of cheapest paths for order w
$r_w = \frac{C_w}{k_w}$	ratio of order cost
$r_{\beta w} = \frac{C_w + \beta_w}{k_w}$	ratio of tolled order costs
$r_{\beta w} = \frac{\sum_{p \in \mathcal{P}_w} C_w^p + \beta_w^p}{k_w}$	ratio of tolled path costs

1 Introduction

The graduation project, on which this report is based, was carried out at TNO, in the department Cyber Security & Robustness. This work has been carried out within the project 'Complexity Methods for Predictive Synchromodality' (Comet-PS), supported by NWO (the Netherlands Organisation for Scientific Research), TKI-Dinalog (Top Consortium Knowledge and Innovation) and the Early Research Program 'Grip on Complexity' of TNO (The Netherlands Organisation for Applied Scientific Research).

In Section 1.1 we will introduce the problem framework as used by TNO to describe the problems in Comet-PS and introduce the term Synchromodality. In Section 1.2 we will introduce the notation used throughout the thesis. In Section 1.3 we specify the research questions and give an outline of the thesis.

1.1. Problem description

The focus of this thesis will be on the User Equilibrium (UE) problem in synchromodal transportation problems, in which we transport containers for multiple customers.
A User Equilibrium is the situation in a transportation network in which all customers are satisfied with the paths they travel by and the corresponding costs they need to pay for transportation. Achieving a User Equilibrium can be an objective when assigning container to transport modalities. Another goal of solving transportation problems can be obtaining a System Optimal (SO) solution, that is a solution in which the total costs of the network are minimized. For a transportation problem, it is not always possible to find a solution that is both SO as well as UE.

We will work from the point of view of a Logistic Service Provider (LSP). The goal of the LSP is to assign containers of multiple customers to modalities. In order to satisfy the customers, the costs of transporting the containers need to be minimized for each customer (UE), but in order to make profit on the transportation, the LSP wants the total costs to be minimized (SO). The LSP has knowledge of all costs in the network, the customer only knows the costs he has to pay to transport his goods on the network.
We will investigate User Equilibria in Min Cost Multi-Commodity Flows (MCMCF) in a synchromodal Space Time Network (STN).

When considering container transport, several transport types can be used [3]. One can choose transporting those by barge, train or truck, which are the most common modalities when transporting containers in the Netherlands.
One type of transport is unimodal transport (see Figure 1.1), in which a container is assigned to one mode for the whole traveling distance.

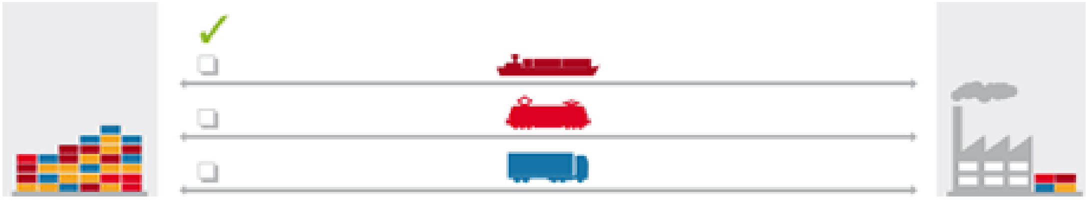


Figure 1.1: Unimodality [3]

Another type is intermodal transport (see Figure 1.2), in which different modalities can be chosen when traveling between different customers and terminals. The last part of the trip is often done by truck (this is also called; last mile by truck).

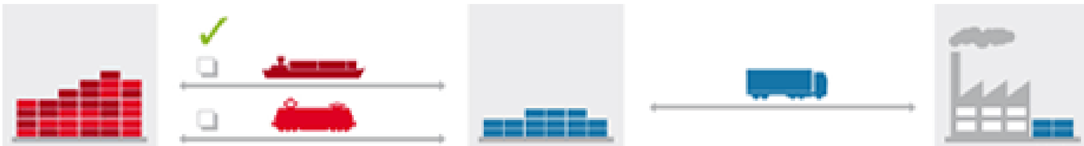


Figure 1.2: Intermodality [3]

A relatively new way of transportation is synchromodal transport, in which at each intermediate terminal, a container can be assigned to a new modality, based on real-time information (see Figure 1.3). In order to provide synchromodal transport, both transporters and customers need to share information about demands and schedules. According to Riessen et al. [20], it is essential for synchromodal transport plans to allow real-time switching, i.e. real-time planning updates, but not many real-time planning methods provide a network-wide plan yet.

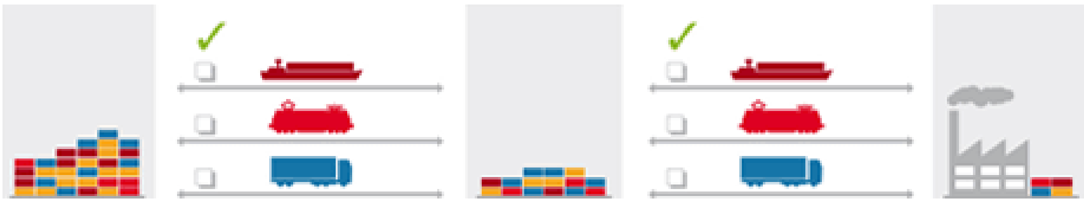


Figure 1.3: Synchromodality [3]

Synchromodality is thus the use of different modalities in one transportation network when transporting units of goods or containers, for which it is possible to switch modalities on every point and time in the network. Containers are assigned to modalities such that it fits best to the wishes of the customer or transporter. In this thesis we will assume that the customers' main requirement is the transport to be as fast and as cheaply as possible. We will take a closer look at optimizing of planning in an agent-centric network from the Logistics Service Providers' point of view, where agent-centric means that satisfying customers is our goal. We assume a customer is satisfied when its containers are transported as cheaply as possible.
Summarized, we work on the Minimum Cost Multi-Commodity Flow problem (MCMCF) in a synchromodal Space Time Network (STN), where our goal is to optimize the cost of all individual customers who want to transport containers. We will define a User Equilibrium on a STN as optimizing all individual customers' objectives. For a definition of an STN we refer to Subsection 1.2.2.

In the synchromodality project at TNO, four problem settings are defined, as can also be seen in Figure 1.4. We divide the transportation problem into those four problem settings, where both infrastructure and events can be fixed or flexible/uncertain. In this project we focus on problem setting 1, to investigate the deterministic problem. In a later stadium of the Comet-PS project, the research can be extended into the more complicated problem settings

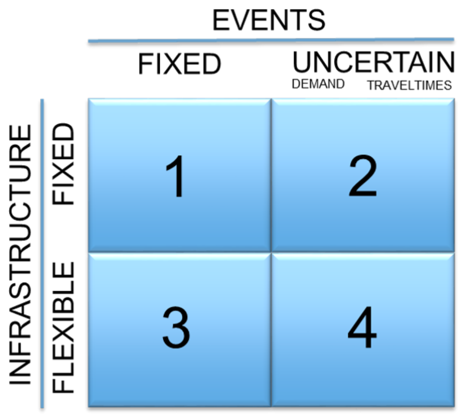


Figure 1.4: Different problem settings

with uncertainties.
Problem setting 1: Assigning containers to modalities, such that the containers reach their destinations, against minimum cost per customer, given that the transportation schedules are fixed and the problem is deterministic.

We will investigate how a User Equilibrium can be obtained on an STN with fixed travel times in problem setting 1. The User Equilibrium problem is investigated already in traffic assignment problems on roads (see Section 2.1), where the travel times of vehicles depend on the occupancy of the roads. We will work on a discrete STN, so we will provide a definition of a User Equilibrium on an STN. In this thesis we will investigate the options of toll pricing to see if we can optimize the total network (SO) on the STN and, in addition, obtain a User Equilibrium.

1.2. Technical introduction

We consider the Min Cost Multi-Commodity Flow problem (MCMCF), in which the goal is to transport containers from multiple orders as cheaply as possible through a network where arcs have weights and may have limited capacity. We can write the MCMCF in an integer linear programming (ILP) form and we will present this ILP in Subsection 1.2.1. The Linear Program we use to solve the User Equilibrium traffic assignment problem will be given in Section 2.1, and in Chapter 4, Chapter 5 and Chapter 6 we consider the Linear Programs for our own User Equilibrium problem.

1.2.1. Definitions and notation used in this thesis

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ consists of a set of nodes $v \in \mathcal{V}$ and a set of directed arcs $a \in \mathcal{A}$. Each arc a is a link between two nodes, an origin node v_1 and an end node v_2: $a = (v_1, v_2)$, along which a container can travel. We use x_a to denote the number of units of flow along arc a. We cannot transport fractions of containers, so the arc flow will always be required to be integer.
An Origin-Destination-pair (OD-pair) w is a pair of two nodes, origin location w_O and destination location w_D, so $w = (w_O, w_D)$, which is not necessarily an arc. The number of containers an order wants to transport from w_O to w_D is denoted by d_w, the demand of order w.
A path p consists of a sequence of adjacent arcs between two nodes. In our problem we only consider paths between origin and destination nodes. f_p denotes the path flow of path p (always integer), with $p \in \mathcal{P}_w$, $w \in \mathcal{W}$, where $\mathcal{P}_w$ is the set of all paths for OD-pair w and $\mathcal{W}$ is the set of all OD-pairs. The total path set is $\mathcal{P} := \bigcup_{w \in W} \mathcal{P}_w$. The costs of an arc a are denoted by c_a and the path costs of path p are denoted by C_w^p or C^p. The capacity of an arc is denoted by m_a and the capacity of a path is denoted by m_p.

In this thesis we use the MCMCF in the following ILP form:

$$\min \sum_{a \in \mathcal{A}} c_a x_a \tag{1.1}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_w} f_p = d_w \ \forall\ w \in \mathcal{W} \tag{1.2}$$

$$\sum_{p \in \mathcal{P}} \delta_{ap} f_p = x_a \ \forall\ a \in \mathcal{A} \tag{1.3}$$

$$x_a \leq m_a \ \forall\ a \in \mathcal{A} \tag{1.4}$$

$$f_p \in \mathbb{N}_0 \ \forall\ p \in \mathcal{P} \tag{1.5}$$

$$x_a \in \mathbb{N}_0 \ \forall\ a \in \mathcal{A} \tag{1.6}$$

with $\delta_{ap} = \begin{cases} 1 & \text{if } a \text{ is contained in } p,\ \forall\ a \in \mathcal{A},\ p \in \mathcal{P}, \\ 0 & \text{otherwise.} \end{cases}$

Objective 1.1 minimizes the total cost made in the network, Constraints 1.2 are the demand constraints, Constraints 1.3 state that the flow on arc a is equal to the sum of the flow on all paths p that contain arc a, Constraints 1.4 are the capacity constraints and Constraints 1.5 and 1.6 assures the non-negativity of all flows. Note that combining Constraints 1.3 and 1.5 make Constraints 1.6 superfluous.

1.2.2. Space Time Networks

A Space Time Network (STN) can be seen as a time-expanded network, as defined as follows:

Definition 1 (Time-expanded network). [22] Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a network with capacities m_a, non-negative integral transit times τ , and costs c_a on the arcs. For a given time horizon $T \in \mathbb{Z}_{>0}$, the corresponding time-expanded network $\mathcal{G}^T = (\mathcal{V}^T, \mathcal{A}^T)$ with capacities and costs on the arcs is defined as follows. For each node $v \in \mathcal{V}$ we create T copies $v_0, v_1, \dots, v_{T-1}$, that is,

$$\mathcal{V}^T := \{v_\theta \mid v \in \mathcal{V}, \theta = 0, 1, \dots, T-1\}.$$

For each arc $a = (v, z) \in \mathcal{A}$, there are $T - \tau_a$ copies $a_0, a_1, \dots, a_{T-1-\tau_e}$ where arc a_θ connects node v_θ to node $z_{\theta+\tau_e}$. Arc a_θ has capacity $m_{a_\theta} := m_a$ and cost $c_{a_\theta} := c_a$. Moreover, $\mathcal{A}^T$ contains holdover arcs $(v_\theta, v_{\theta+1})$ for $v \in \mathcal{V}$ and $\theta = 0, \dots, T-2$. The capacity of holdover arcs is infinite and they have zero cost. Summarizing, the set of arcs $\mathcal{A}^T$ is given by

$$\mathcal{A}^T := \left\{a_\theta = \left(v_\theta, z_{\theta+\tau_a}\right) \mid a = (v, z) \in \mathcal{A},\ \theta = 0, 1, \dots, T-1-\tau_a\right\}$$
$$\bigcup \ \{(v_\theta, v_{\theta+1}) \mid v \in \mathcal{V},\ \theta = 0, 1, \dots, T-2\}.$$

We now show an example of the transition of a regular graph into an STN:

Example 1. Given is the graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ as in Figure 1.5 with $\mathcal{V} = \{s, v, w, e\}$ being the set of nodes and $\mathcal{A} = \{(s, v), (s, w), (v, w), (v, e), (w, e)\}$ being the set of directed arcs.

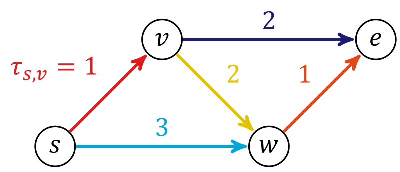


Figure 1.5: Static network $\mathcal{G}$

The transit times τ_a on the arcs are given in Figure 1.5. There are no capacity constraints. In this example we say the costs are equal to the transit times: $c_a = \tau_a\ \forall\ a \in \mathcal{A}$.

To transform this network to an STN, we first take $T = 6$. Then we can create $\mathcal{G}^6 = (\mathcal{V}^6, \mathcal{A}^6)$, with

$$
\begin{aligned}
\mathcal{V}^6 = &\{s_0, s_1, \ldots, s_5, s_6,\ v_0, v_1, \ldots, v_5, v_6,\ w_0, w_1, \ldots, w_5, w_6,\ e_0, e_1, \ldots, e_5, e_6\},\\
\mathcal{A}^6 = &\{(s_0,v_1),(s_1,v_2),(s_2,v_3),(s_3,v_4),(s_4,v_5),(s_5,v_6),\ (s_0,w_3),(s_1,w_4),(s_2,w_5),(s_3,w_6),\\
&(v_0,w_2),(v_1,w_3),(v_2,w_4),(v_3,w_5),(v_4,w_6),\ (v_0,e_2),(v_1,e_3),(v_2,e_4),(v_3,e_5),(v_4,e_6),\\
&(w_0,e_1),(w_1,e_2),(w_2,e_3),(w_3,e_4),(w_4,e_5),(w_5,e_6),\ (s_0,s_1),(s_1,s_2),(s_2,s_3),(s_3,s_4),(s_4,s_5),(s_5,s_6),\\
&(v_0,v_1),(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_4,v_5),(v_5,v_6),\ (w_0,w_1),(w_1,w_2),(w_2,w_3),(w_3,w_4),(w_4,w_5),(w_5,w_6),\\
&(e_0,e_1),(e_1,e_2),(e_2,e_3),(e_3,e_4),(e_4,e_5),(e_5,e_6)\}
\end{aligned}
$$

The set of arcs $\mathcal{A}^6$ can be found graphically in Figure 1.6. We see that for every arc in $\mathcal{A}$ copies are made per time step. △

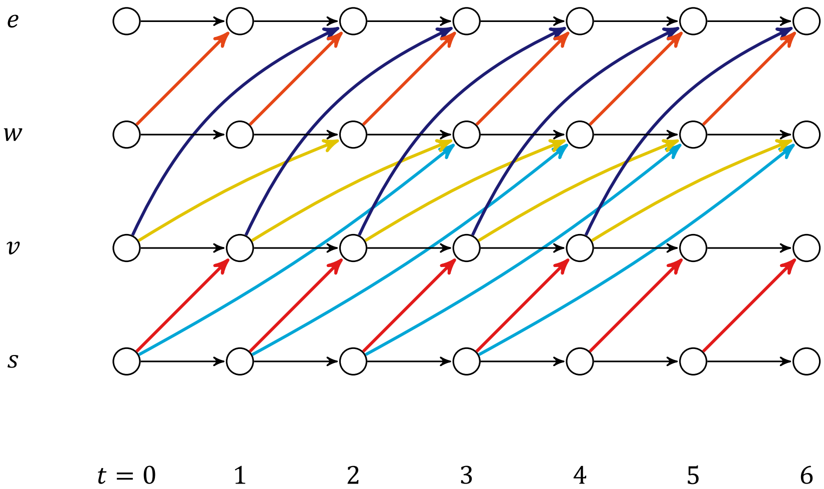


Figure 1.6: STN of $\mathcal{G}$: $\mathcal{G}^6$

In this thesis we will use fixed traveling schemes, so departure and arrival times of modalities are known in advance, as well as the release time of orders. Traveling arcs between different locations represent barges, trains or trucks which are already planned, and we only need to assign containers to certain modalities regarding the capacities of those modalities. In Figure 1.7 for example, we see three locations l: $\mathcal{V} = \{1,2,3\}$ and four traveling arcs: $\mathcal{A} = \{a,b,c,d\}$. Here the number of time steps is $T = 5$. When an order wants to travel from location $l = 1$ to $l = 2$, it has the choice between arc a and b, with travel time 2 and 4 respectively. When an order wants to travel from location $l = 1$ to $l = 3$, there are three path choices: ac, ad and bd, with costs 4, 5 and 5 respectively.

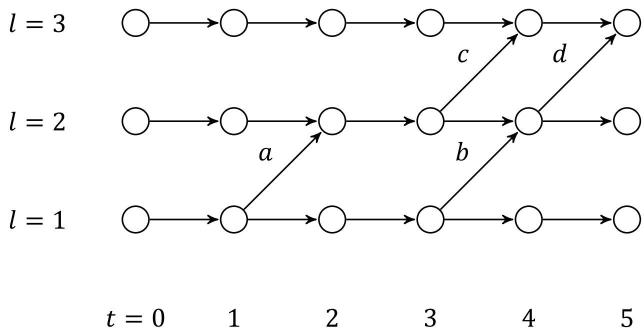


Figure 1.7: Example of an STN

1.3. Thesis structure

The goal of this thesis is to find a method to find a solution of assigning containers to modes in a synchromodal STN, where the solution is both System Optimal as well as User Equilibrium. As part of this research, we also need to define when a User Equilibrium is reached in our problem setting 1, and this definition is different from the definition as used in the literature. We will therefore first investigate how the User Equilibrium problem is defined and solved in the literature. Next, we will define the User Equilibrium on an STN and compare SO and UE solutions on an STN. The following step is to obtain solutions which are both SO and UE, by applying tolls on the STN.

In Chapter 2 we will introduce the User Equilibria as used in the literature, and the application of the Lagrangian Dual on User Equilibria in traffic assignment problems. We will also discuss the use of tolls on traffic problems in the literature. In Chapter 3 we introduce our use of the Space Time Network and we give the definition of the User Equilibrium in our problem setting. In this chapter we will also show how we find System Optimal and User Equilibrium solutions in Space Time Networks. Then in Chapter 4 we show our method to find a solution of assigning containers on a Space Time Network which is both SO as well as UE, by using tolls on a path-based level. In Chapter 5 we do this by using tolls on a order-based level. In Chapter 6 we will combine the previous methods to obtain “fair” path-based toll solutions. In Chapter 7 we state our conclusions and propose future work.

2

User Equilibrium and toll systems in the literature

In Section 2.1 we describe how User Equilibrium is applied in the literature and in Section 2.2 we give an outline of tolls described in the literature.

2.1. User Equilibrium

In this thesis we will look for a User Equilibrium (UE) solution in discrete time transport networks, especially in a Space Time Network (STN). To introduce the term User Equilibrium in an STN, we will first look how a UE in traffic assignment problems (TAP) is defined. With the TAP we mean the problems in which units, which can be goods, containers, but also passenger traffic with cars or public transport, are transported on networks with travel times depending on the occupancy of links, such as roads.

A User Equilibrium means intuitively that every order which has to be transported on a given network, is scheduled such that it is handled the best way possible, which means the fastest, the cheapest, depending on what the customer finds most important. But unfortunately that is hardly ever possible, because in most networks capacity constraints make it impossible to handle all orders the best way possible.

We will now introduce the notions of equilibrium stated by Wardrop [27]. In a traffic assignment problem on a transport network, where the travel time depends on the occupancy of roads, the following criterion denotes a User Equilibrium (UE) (we will give a definition for UE as used in the literature in Definition 3 on page 11): The journey times on all the routes actually used are equal, and less than those that would be experienced by a single vehicle on any unused route. Here a route is a path between an origin and destination location.
In this literature review, the given criterion for a UE is valid. Beside the definition of a User Equilibrium, Wardrop also defines a System Optimal solution.

Definition 2 (System Optimal (SO))**.** The average journey time is a minimum.

We mean by System Optimal that the total cost (or total travel time) of the problem is minimized, regardless of the conditions for the individual customers.

We give a brief outline of User Equilibria as found in the literature in Subsection 2.1.1 and we apply this on a small example network to see what problems occur when solving UE problems. In Subsection 2.1.2 we take a closer look at the objective function as used in most of the literature about UE and in Subsection 2.1.3 we compare the objective functions of User Equilibrium and System Optimal in the literature. In Subsection 2.1.4 we briefly discuss the existence of a unique path flow in User Equilibria and in Subsection 2.1.5 and in Subsection 2.1.6 we explain how the Lagrangian Dual is used to solve User Equilibrium problems.

2.1.1. User Equilibrium in the literature

Most of the literature about User Equilibrium is based on network congestion, where travel times on roads depend on occupancy of traveling arcs, as in traffic assignment problems. The literature review in this section is about User Equilibria on those type of networks, our own definition of User Equilibria will be given in Chapter 3.

Van Essen et al. [24] give a proper review of ways to force a UE into an SO by diffusing travel information to stimulating some agents to travel non-selfishly to achieve cheaper total costs.
Peeta and Mahmassani [19] investigate both the System Optimal and the User Equilibrium Time-Dependent Traffic Assignment. They show that the more goods have to be transported, the more the solutions of the two models differ from each other. Bar-Gera [4] provide a solution method for the UE traffic assignment problem which is computationally efficient, memory conserving and an origin-based solution method.
Xu et al. [29] proposes a stochastic User Equilibrium for a passenger transport network.

Miyagi et al. [18] consider a traffic assignment problem from the view of game theory. They assume drivers have knowledge of the network and a Nash Equilibrium (which corresponds to a User Equilibrium) is achievable. Wagner [25] shows that the existence of a Nash Equilibrium is guaranteed under some natural assumptions on the travel time models. Also Wang and Yang [26] show the equality of Nash Equilibrium and User Equilibrium. Levy et al. [16] consider selfish agents in a traffic assignment problems, and apply properties of game theory on traffic problems. They start from finding User Equilibrium solution, in which all agents take the best route for themselves, based on their route choice experiences in the past. The question then is if it is possible to obtain a System Optimal solution, in which agents are still selfish.

The relationship between the User Equilibrium and the System Optimal can be examined by the Price of Anarchy [21], a system often used in both economics and game theory, that measures how the efficiency of a system degrades due to selfish behavior of its customers. Roughgarden considers the Price of Anarchy ρ, which compares the total costs of the network when solved as an SO to the UE solution: $\rho = \frac{z_{UE}}{z_{SO}}$, where z_{UE} and z_{SO} denote the total cost value of the UE solution and the SO solution, respectively.

Bar-Gera et al. [5] consider the User Equilibrium problem with the focus on spreading flow over the network (not time-dependent). They also introduce several criteria which can be taken into consideration for choosing UE solution methods. Their most important addition to the subject is the condition of proportionality: the same proportions apply to all travelers facing a choice between a pair of alternative paths, regardless of their origins and destinations. This means that if a link in the network is used by multiple OD-pairs, the amount of flow on that link belonging to one OD-pair compared to the total link flow, is proportional to the total OD flow compared to the total flow on the network. An important implication of proportionality is that any route that can be used under the UE conditions, will be used.

Corman et al.[8] consider the application of multimodal transport to provide a User Equilibrium solution, with the choice of modality based on the wishes of the customers. They assume customers have access to a system for publishing demand and offering transportation possibilities. Moreover, they assume everybody has access to truck transportation, so transport is always possible, regardless of the fact that other modalities are not available. They define every customer as one unit of transport, which has to choose one specific mode for the whole travel distance. The goal is to assign customers to modes in such a way that no customer will change its departure time and its route (and thus will not change its mode), to provide all customers a sufficient route.

Borchers et al.[7], give the following definition and lemma of a User Equilibrium in network with travel times based on occupancies of the links:

Definition 3. [Wardrop equilibrium] Consider a traffic flow $(\boldsymbol{f},\boldsymbol{x})$ with corresponding induced costs $c_a(\boldsymbol{x})$. Then $(\boldsymbol{f},\boldsymbol{x})$ is called a Wardrop equilibrium (or User Equilibrium) if $\forall$ paths $r, q, \in \mathcal{P}_w$, and $\forall$ demands $w \in \mathcal{W}$, the following condition is satisfied:

$$f_r > 0 \Rightarrow \begin{cases} C^r(\boldsymbol{x}) = C^q(\boldsymbol{x}) & \text{if } f_q > 0 \\ C^r(\boldsymbol{x}) \leq C^q(\boldsymbol{x}) & \text{if } f_q = 0 \end{cases}$$

Lemma 1. *The following are equivalent for a traffic flow $(\boldsymbol{f},\boldsymbol{x})$:*

1. *The flow $(\boldsymbol{f},\boldsymbol{x})$ is a Wardrop equilibrium,*

2. *Beckmann's formulation:*
   $(\boldsymbol{f},\boldsymbol{x})$ is a solution of the (convex) program:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{f}} z(\boldsymbol{x}) := \min_{\boldsymbol{x}} & \sum_{a\in\mathcal{A}} \int_0^{x_a} c_a(\tau)d\tau \\ \textit{s.t.} & \sum_{p\in\mathcal{P}_w} f_p = d_w \ \forall\, w \in \mathcal{W} \\ & \sum_{p\in\mathcal{P}} \delta_{ap} f_p = x_a \ \forall\, a \in \mathcal{A} \\ & f_p \in \mathbb{N}_0 \ \forall\, p \in \mathcal{P} \end{aligned}$$

$$\text{with } \delta_{ap} = \begin{cases} 1 & \textit{if } a \textit{ is contained in } p,\ \forall\, a \in \mathcal{A},\ p \in \mathcal{P}, \\ 0 & \textit{otherwise.} \end{cases}$$

The correctness of the lemma is proven in Subsection 2.1.2.

We will now show an example of a traffic network with travel times depending on the occupancy of the network, where we use the Beckmann formulation of Lemma 1.

User Equilibrium in the literature: an example

Example 2. In this example there are two orders, namely order 1 with demand $d_1 = 10$ traveling from A to B and order 2 with demand $d_2 = 6$ from A to C.

The network in which orders have to be transfered is given in Figure 2.1, where each directed arc has a cost function $c_a(x_a) = s_a + x_a$ (the cost function c_a represents the travel time on arc a in this example), where the cost function consists of two parts: s_a the standard cost and x_a the occupancy on the arc. The cost functions are given in Figure 2.1. To solve this User Equilibrium problem, we have the

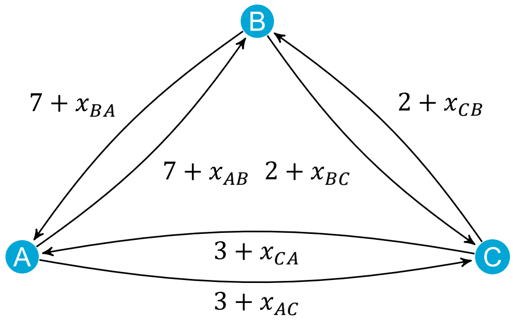


Figure 2.1: Example of a UE problem

following objective function and constraints as formulated by Beckmann (see Lemma 1):

$$\min \sum_{a \in \mathcal{A}} \int_0^{x_a} c_a(\tau) d\tau \tag{2.1}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_1} f_p = 10, \ \text{ with } \ \mathcal{P}_1 = \{AB,\ ACB\} \tag{2.2}$$

$$\sum_{p \in \mathcal{P}_2} f_p = 6, \ \text{ with } \ \mathcal{P}_2 = \{ABC,\ AC\} \tag{2.3}$$

$$\sum_{p \in \mathcal{P}} \delta_{ap} f_p = x_a \ \forall \ a \in \mathcal{A} \tag{2.4}$$

$$f_p \in \mathbb{N}_0 \ \forall \ p \in \mathcal{P} \tag{2.5}$$

with $\delta_{ap} = \begin{cases} 1 & \text{if } a \text{ is contained in } p, \ \forall \ a \in \mathcal{A},\ p \in \mathcal{P}, \\ 0 & \text{otherwise.} \end{cases}$

Expression 2.1 is the objective function of the User Equilibrium, Constraint 2.2 is the demand constraint of order 1 with demand $d_1 = 10$ and Constraint 2.3 the demand constraint of order 2 with $d_2 = 6$. Constraint 2.4 states that the flow on arc a is the sum of all paths p that contain arc a, and Constraint 2.5 assures non-negativity of all (path) flows (which implies non-negativity of arc flows). Note that in this example there are no capacity constraints.
In this network the objective function is

$$\sum_{a \in \mathcal{A}} \int_0^{x_a} c_a(\tau) d\tau = \min \sum_{a \in \mathcal{A}} \left(s_a x_a + \frac{1}{2} x_a^2 \right).$$

A feasible path flow solution that minimizes the objective function is

$$f_{AB} = 8,\ f_{ACB} = 2,\ f_{AC} = 6,$$

from which the arc flow solution follows

$$x_{AB} = 8,\ x_{AC} = 8,\ x_{CB} = 2.$$

The path costs for the used paths are:

$$\begin{aligned} C_1^{AB} &= 7 + x_{AB} &&= 7 + 8 &&= 15, \\ C_1^{ACB} &= 3 + x_{AC} + 2 + x_{CB} &&= 3 + 8 + 2 + 2 &&= 15, \\ C_2^{AC} &= 3 + x_{AC} &&= 3 + 8 &&= 11. \end{aligned}$$

We see that the path costs for all containers of order 1 are equal. The path cost for the unused path ABC is

$$C_2^{ABC} = 7 + x_{AB} + 2 + x_{BC} = 7 + 8 + 2 + 0 = 17$$

thus f_{AC} is indeed the cheapest route for order 2. This solution has total cost 216.
How do we interpret this path cost? Every individual customer in the network has to pay for the paths used, where the path cost is the sum of arc costs when those arcs are in the path. When multiple containers are using the same arc, the arc cost per container is dependent on all flow put on that arc: the more containers are on the arc, the higher the cost per player on that arc.
The found solution is a UE, but is it SO as well?
When we want to find the SO solution, we can first work out the problem:

$$\begin{aligned}
\min \sum_{a \in \mathcal{A}} c_a(x_a)x_a = \min \sum_{a \in \mathcal{A}} s_a x_a + (x_a)^2 & \\
\text{s.t. } & f_{AB} + f_{ACB} = 10 \\
& f_{AC} + f_{ABC} = 6 \\
& \sum_{p \in \mathcal{P}} \delta_{ap} f_p = x_a \ \forall\ a \in \mathcal{A} \\
& f_p \in \mathbb{N}_0 \ \forall\ p \in \mathcal{P}
\end{aligned}$$

with $\delta_{ap} = \begin{cases} 1 & \text{if } a \text{ is contained in } p,\ \forall\ a \in \mathcal{A},\ p \in \mathcal{P}, \\ 0 & \text{otherwise.} \end{cases}$
When solving this Integer Programming problem, we obtain the same solution as we obtained for the UE objective function. So in this case the SO solution is equal to the UE solution. △

2.1.2. Objective function

As an objective function for User Equilibrium problems, the following expression is often used in the literature:

$$\min \sum_{a \in \mathcal{A}} \int_0^{x_a} c_a(\tau) d\tau$$

Beckmann et al. [6] show the correctness of this objective function when observing UE problems, and is approach for showing that is explained below, using our own notation.

Correctness of the objective and Equilibrium conditions

Beckmann first formulates the Equilibrium Conditions. Given arcs $a \in \mathcal{A}$ and Origin-Destination pairs $w \in \mathcal{W}$, we have the path flow and arc flow variables

$$f_p \geq 0 \ \forall\ p \in \mathcal{P}, \tag{2.6}$$

$$x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \ \forall\ a \in \mathcal{A}, \tag{2.7}$$

with parameters $\delta_{ap} = \begin{cases} 1 & \text{if } a \text{ is contained in } p,\ \forall\ a \in \mathcal{A},\ p \in \mathcal{P}, \\ 0 & \text{otherwise.} \end{cases}$
From Constraints 2.6 and 2.7 follow the non-negativity constraint for arcs:

$$x_a \geq 0 \ \forall\ a \in \mathcal{A}.$$

Beckmann also states that the total sum of path flow per order equals the demand:

$$d_w = \sum_{p \in \mathcal{P}_w} f_p,$$

with d_w the number of units of demand w, originated at location $v = w_O$. f_p is the path flow on path p with w_O as the begin point and w_D as the end point of the path. Because there are multiple paths possible for order w, the demand equals the sum of flow on all possible paths belonging to that order.

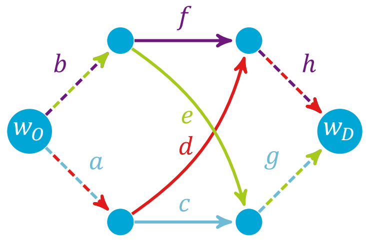


Figure 2.2: OD-pair w with paths $p_1 = acg$, $p_2 = adh$, $p_3 = beg$, $p_4 = bfh$

In Figure 2.2 an example of an OD pair is given, with all optional paths for order w: $\mathcal{P}_w = \{p_1, p_2, p_3, p_4\}$.

In case the cost functions are constant functions and there are no capacity constraints ($m_a = \infty \ \forall \ a \in \mathcal{A}$), there is only one cheapest route, and hence

$$d_w = \min_{\substack{p \in \mathcal{P}_w: \\ \sum_{p \in \mathcal{P}} C^p \text{ is minimized}}} f_p \ \forall \ w \in \mathcal{W}.$$

Define c_a the arc cost on arc $a \in \mathcal{A}$ and $C^{(v, w_D)}$ the minimum path cost from node v to w_D. $f_{(v_i, w_D)}$ is thus defined as the path flow for the cheapest path between node v_i and destination w_D.

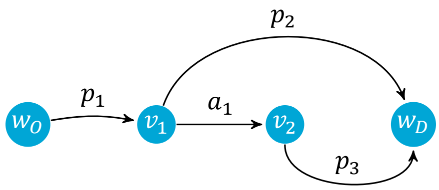


Figure 2.3: OD-pair w with paths p_1, p_2, p_3 and arc a_1. Path costs $C_w^{(v_1, w_D)} = \min\{C_w^{p_2}, C_w^{a_1 p_3}\}$, $C_w^{(v_2, w_D)} = C_w^{p_3}$. Arc cost $c_{(v_1, v_2)} = c_{a_1}$.

Take for example the graph in Figure 2.3. Arc a_1 connects v_2 to v_1, $a_1 = (v_1, v_2)$. Then with this arc a_1 we can extend the path from v_2 to w_D (denoted by p_3 to a path from v_1 to w_D. There are thus two optional paths between v_1 and w_D, namely p_2 and $a_1 p_3$, but the graph does not state whihc is the cheapest route. We can define the minimum path cost of path (v_1, w_D) as:

$$C_w^{(v_1, w_D)} = \min\{C_w^{p_2}, C_w^{a_1 p_3}\} = \min\{C_w^{p_2}, c_{a_1} + C_w^{p_3}\} = \min\left\{C_w^{p_2}, c_{a_1} + C_w^{(v_2, w_D)}\right\},$$

Which implies

$$C_w^{(v_1, w_D)} \leq c_{a_1} + C_w^{(v_2, w_D)}.$$

We can rewrite this result: for every pair of points $v_1, v_n \in \mathcal{V}$ there exists a uniquely determined number $C^{(v_1, v_n)}$ such that for all $a = (v_1, v_2)$ with v_2 an adjacent node of v_1:

$$C_w^{(v_1, v_n)} - C_w^{(v_2, v_n)} \leq c_a \text{ with equality for some } v_2; \ C_w^{(v_n, v_n)} = 0. \tag{2.8}$$

The converse is also true: if for every pair of points $v_1, v_n \in \mathcal{V}$ there is a unique value $C_w^{(v_1, v_n)}$ satisfying (2.8), then $f_{(v_1, v_n)}$ represents the flow on the cheapest path with path cost:

$$C_w^{(v_1, v_n)} := \min_{v_2, v_3, \ldots v_{n-2}, v_{n-1}} \left(c_{(v_1, v_2)} + c_{(v_2, v_3)} + \cdots + c_{(v_{n-2}, v_{n-1})} + c_{(v_{n-1}, v_n)}\right).$$

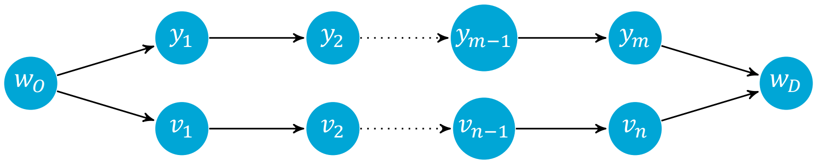


Figure 2.4: OD-pair w with 2 paths.

As an example we show in Figure 2.4 an OD pair $w = (w_O, w_D)$ with two optional paths, where

$C_w^{w_O,w_D} = \min\{(c_{w_O,v_1} + c_{(v_1,v_2)} + \cdots + c_{(v_{n-1},v_n)} + c_{(v_n,w_D)}), (c_{w_O,y_1} + c_{(y_1,y_2)} + \cdots + c_{(y_{m-1},v_n)} + c_{(y_m,w_D)})\}.$

Now write the number of trips for OD-pair w on path q (thus path flow f_q) as a function of the path cost: $f_q = l_q(C^q)$, with the inverse relationship $C^q = g_q(f_q)$. l_q is a monotonic (decreasing) function (the more path q costs, the less often that path is used), so g_q exists. We may therefore assume

$$\frac{\partial g_q}{\partial f_q} < 0, \tag{2.9}$$

wherever g_q is differentiable. $c_a = h_a(x_a)$ shows the relation between arc cost and arc flow, which is never decreasing:

$$\frac{\partial h_a}{\partial x_a} \geq 0. \tag{2.10}$$

We can say that for arcs that are not in a cheapest route to a location w_D, the flow related to that location on that arc is zero:

$$\delta_{ap} f_p = 0 \text{ if } C^{(v_1,w_D)} - C^{(v_2,w_D)} < c_a, \text{ with } a = (v_1, v_2),\ \forall\ p \in \mathcal{P}_w\ \forall\ (w_O, w_D) \in \mathcal{W}.$$

We can write:

$$C^{(v_1,w_D)} - C^{(v_2,w_D)} \begin{Bmatrix} = \\ \leq \end{Bmatrix} c_a \quad \text{if} \quad \delta_{ap} f_p \begin{Bmatrix} > \\ = \end{Bmatrix} 0.$$

Substitution of the cost and flow variables gives us the following system:

$$g_{(v_1,w_D)}\left(f_{(v_1,w_D)}\right) - g_{(v_2,w_D)}\left(f_{(v_2,w_D)}\right) \begin{Bmatrix} = \\ \leq \end{Bmatrix} h_a(x_a) \quad \text{if} \quad \delta_{ap} f_p \begin{Bmatrix} > \\ = \end{Bmatrix} 0. \tag{2.11}$$

Multiplication on both sides by $\delta_{ap} f_p$ with arc $a = (v_1, v_2)$ and addition yields:

$$\sum_{(v_1,v_2)=a\in\mathcal{A}} \sum_{p\in\mathcal{P}} \left[g_{(v_1,w_D)}\left(f_{(v_1,w_D)}\right) - g_{(v_2,w_D)}\left(f_{(v_2,w_D)}\right)\right] \delta_{ap} f_p = \sum_{(v_1,v_2)=a\in\mathcal{A}} \sum_{p\in\mathcal{P}} h_a(x_a) \cdot \delta_{ap} f_p.$$

We can rewrite both sides of this equation to find the expression with $x_a = \sum_{p\in\mathcal{P}} \delta_{ap} f_p\ \forall\ a \in \mathcal{A}$:

First we rewrite the right hand side:

$$\sum_{a\in\mathcal{A}} \sum_{p\in\mathcal{P}} h_a(x_a) \cdot \delta_{ap} f_p = \sum_{a\in\mathcal{A}} h_a(x_a) \sum_{p\in\mathcal{P}} \delta_{ap} f_p = \sum_{a\in\mathcal{A}} h_a(x_a) x_a.$$

Now the left hand side:

$$\sum_{a\in\mathcal{A}}\sum_{p\in\mathcal{P}}\left[g_{(v_1,w_D)}\left(f_{(v_1,w_D)}\right)-g_{(v_2,w_D)}\left(f_{(v_2,w_D)}\right)\right]\delta_{ap}f_p$$
$$=\sum_{p\in\mathcal{P}}\left[g_{(v_1,w_D)}\left(f_{(v_1,w_D)}\right)-g_{(v_2,w_D)}\left(f_{(v_2,w_D)}\right)+g_{(v_2,w_D)}\left(f_{(v_2,w_D)}\right)-g_{(v_3,w_D)}\left(f_{(v_3,w_D)}\right)+\dots\right.$$
$$\left.+g_{(v_{n-1},w_D)}\left(f_{(v_{n-1},w_D)}\right)-g_{(v_n,w_D)}\left(f_{(v_n,w_D)}\right)\right]f_p$$
[use $v_n = w_D \Rightarrow \ g_{(v_n,w_D)} = 0$]
$$=\sum_{p\in\mathcal{P}}g_{(v_1,w_D)}\left(f_{(v_1,w_D)}\right)f_p$$
$$=\sum_{p\in\mathcal{P}}g_p(f_p)f_p$$

So we can conclude

$$\sum_{p\in\mathcal{P}}g_p(f_p)f_p=\sum_{a\in\mathcal{A}}h_a(x_a)x_a,$$
$$\Leftrightarrow\quad\sum_{p\in\mathcal{P}}C^pf_p=\sum_{a\in\mathcal{A}}c_ax_a, \tag{2.12}$$

with the left-hand side representing the sum of all route costs and the right-hand side is the sum of all arc costs.

The following step Beckmann takes, is showing the existence of solutions to the Equilibrium Conditions. Consider the function

$$H(\dots,\delta_{ap}f_p,\dots)=\sum_{p\in\mathcal{P}}\int_0^{f_p}g_p(\tau)d\tau-\sum_{a\in\mathcal{A}}\int_0^{x_a}h_a(\tau)d\tau, \tag{2.13}$$

which we want tho maximize. Differentiate Equation 2.13 with respect to $\delta_{ap}f_p$ after substituting $x_a=\sum_{p\in\mathcal{P}}\delta_{ap}f_p$ from Equation 2.7. This gives us

$$\frac{\partial H}{\partial(\delta_{ap}f_p)}=g_{(v_1,w_D)}\left(f_{(v_1,w_D)}\right)-g_{(v_2,w_D)}\left(f_{(v_2,w_D)}\right)-h_a(x_a).$$

The objective is to maximize H. To have a maximum, all derivatives of H in the region with $x_a>0\ \forall\ a\in\mathcal{A}$, must be less than or equal to zero. From (2.11) we find:

$$\frac{\partial H}{\partial(\delta_{ap}f_p)}=0\text{ if }\delta_{ap}f_p>0\text{ and }\frac{\partial H}{\partial(\delta_{ap}f_p)}\leq 0\text{ if }(\delta_{ap}f_p)=0. \tag{2.14}$$

From Equation 2.10 follows that

$$\lim_{x_a\to m_a}h_a(x_a)=\infty. \tag{2.15}$$

In the closed set defined by the two conditions; $0\leq\delta_{ap}f_p$ and $0\leq x_a\leq k_a<m_a$ for some suitable k_a, the function $H(\dots,\delta_{ap}f_p,\dots)$ is continuous since it is a sum of indefinite integrals of Riemann integrable functions. Beckmann concludes that H assumes its maximum at some point. This will only be the case when k_a it not too close to m_a because of Equation 2.15. At the point of maximum the necessary conditions Equation 2.13 must be satisfied. But these conditions are identical to those in (2.11). Therefore there must exist a solution to the inequalities in (2.11) and the supplementary defining Equation 2.7. H was constructed in such a way that the first order conditions for a maximum (2.14) become identical with the equilibrium condition (2.11).

Beckmann shows that finding a solution for this objective function

$$\max\ H(\dots,\delta_{ap}f_p,\dots) = \max \sum_{p\in\mathcal{P}} \int_0^{f_p} g_p(\tau)d\tau - \sum_{a\in\mathcal{A}} \int_0^{x_a} h_a(\tau)d\tau$$

is equivalent to finding an equilibrium that satisfies the following conditions:

$$C^{(v_1,w_D)} - C^{(v_2,w_D)} \begin{Bmatrix} = \\ \leq \end{Bmatrix} c_a \quad \text{if} \quad \delta_{ap}f_p \begin{Bmatrix} > \\ = \end{Bmatrix} 0, \text{ with } a = (v_1, v_2).$$

Why does this relation prove the fact that we are allowed to use the following objective function?

$$\min \sum_{a\in\mathcal{A}} \int_0^{x_a} c_a(\tau)d\tau$$

(Note that $c_a(x_a)$ here is equal to the function $h_a(x_a)$).

Given the function H from Beckmann, we can easily rewrite the objective:

$$\begin{aligned}\max\ H &= \max \sum_{p\in\mathcal{P}} \int_0^{f_p} g_p(\tau)d\tau - \sum_{a\in\mathcal{A}} \int_0^{x_a} h_a(\tau)d\tau \\ &= \min \sum_{a\in\mathcal{A}} \int_0^{x_a} h_a(\tau)d\tau - \sum_{p\in\mathcal{P}} \int_0^{f_p} g_p(\tau)d\tau\end{aligned}$$

This objective aims to minimize the difference between the sum of arc costs and path costs, with h_a a non-decreasing function (see Equation 2.10) and g_p is a decreasing function (see (2.9)).

2.1.3. User Equilibrium vs. System Optimal

In general, the total cost of an SO solution is lower than the total cost of a UE solution. However, we can prove that the costs of SO and UE solutions are similar in the case that all arc cost functions are constant, given the definition of a UE as used in the literature. We prove this as follows, with z_{SO} and z_{UE} the objective values for the SO and UE objective functions, respectively:

$$z_{SO} = \sum_{a\in\mathcal{A}} x_a \cdot c_a(x_a) = \sum_{a\in\mathcal{A}} x_a \cdot c_a = \sum_{a\in\mathcal{A}} \int_0^{x_a} c_a d\tau = \sum_{a\in\mathcal{A}} \int_0^{x_a} c_a(\tau)d\tau \ = z_{UE}.$$

The problem in this case is, that in a graph with capacity constraints, an SO solution can be highly adverse for some player in the network, because that player is forced to take an expensive route for the network to be of optimal cost, while another player can take its cheapest route. But with constant arc costs, this solution is called UE as well, but it certainly is not what we want in a UE. So we will give our own definition of a User Equilibrium in Chapter 3. We will show in the next two examples which solutions can be found in a network with constant arc cost functions and capacity constraints, and which UE and SO solutions occur.

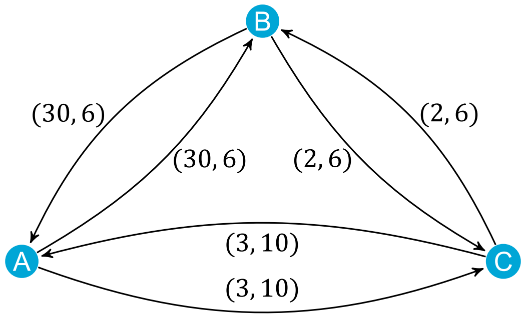


Figure 2.5: Example 3, (c_a, m_a) the cost and capacity for arc a, respectively.

Example 3. In this example we have two orders, order 1 traveling from A to B with demand $d_1 = 10$ and order 2 traveling from A to C with demand $d_2 = 6$. The network is given in Figure 2.5.

We have the following ILP:

$$\min \sum_{a \in \mathcal{A}} c_a x_a \tag{2.16}$$
$$f_{AB} + f_{ACB} = 10 \tag{2.17}$$
$$f_{AC} + f_{ABC} = 6 \tag{2.18}$$
$$\sum_{p \in \mathcal{P}} \delta_{ap} f_p = x_a \ \forall\ a \in \mathcal{A} \tag{2.19}$$
$$x_a \le m_a \ \forall\ a \in \mathcal{A} \tag{2.20}$$
$$f_p \in \mathbb{N}_0 \ \forall\ p \in \mathcal{P} \tag{2.21}$$

Note that the objective in (2.16) is equal for both the SO and UE problem, because of the constant arc costs. Constraints 2.17 and 2.18 denote the demand constraints, Constraint 2.19 is the arc path flow conservation, Constraint 2.20 is the capacity constraint for all arcs and Constraint 2.21 make sure all flows are non-negative and integer.

The arc flow solution of the problem above is

$$x_{AB} = 6,\ x_{AC} = 10,\ x_{CB} = 4,$$

and thus the path flow solution is

$$f_{AB} = 6,\ f_{ACB} = 4,\ f_{AC} = 6.$$

Recall that this solution is both SO and UE, because we are using the definitions found in the literature. The path flows have the following, path costs

$$C_1^{AB} = 30,\ C_1^{ACB} = 5,\ C_2^{AC} = 3$$

which gives a total cost of 218 and the total costs per customer:
$C_1 = 200$, thus an average of cost 20 per container, and for $C_2 = 18$, an average of cost 3 per container.

We see that order 1, which travels from A to B, is in disadvantage. Suppose we let the customers choose their paths themselves, and we let order 1 decide first, then we obtain the following solution:

$$x_{AB} = 6,\ x_{AC} = 10,\ x_{BC} = 2, x_{CB} = 6 \text{ and } f_{AB} = 4,\ f_{ACB} = 6,\ f_{AC} = 4,\ f_{ABC} = 2,$$

with corresponding path costs

$$C_1^{AB} = 30,\ C_1^{ACB} = 5,\ C_2^{AC} = 3,\ C_2^{ABC} = 32,$$

which gives a total cost of 226 and the total costs per customer: $C_2 = 76$, thus an average of cost $12\frac{2}{3}$ per container, and for $C_1 = 150$, an average of cost 15 per container.
The cheapest path for order 1 is ACB with cost 5 per order and for order 2 it is the direct arc AC with cost 3. However, due to the capacity constraints, for $d_1 = 10$, it is not possible for all containers of order 1 to transport them via path ACB and thus the more expensive path AB needs to be used. According to the definitions of UE and SO as given in the literature, this solution is not UE nor SO.

△

More interesting is the following example:

Example 4. We use the same network as in the previous example, but we adjust the capacity constraints. The network is given in Figure 2.6. Again we have two orders, order 1 traveling from A to B with demand $d_1 = 10$ and order 2 traveling from A to C with demand $d_2 = 20$.

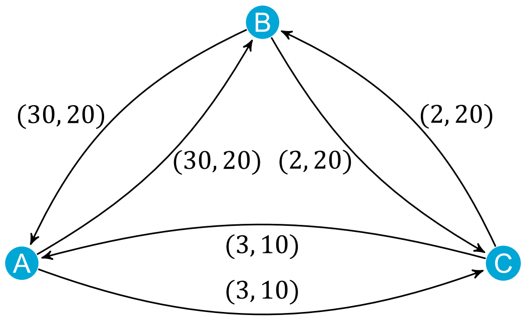


Figure 2.6: Example 4, (c_a, m_a) the cost and capacity for arc a, respectively.

We have the following ILP:

$$\min \sum_{a\in\mathcal{A}} c_a x_a$$
$$f_{AB} + f_{ACB} = 10$$
$$f_{AC} + f_{ABC} = 20$$
$$\sum_{p\in\mathcal{P}} \delta_{ap} f_p = x_a \ \forall \ a \in \mathcal{A}$$
$$x_a \leq m_a \ \forall \ a \in \mathcal{A}$$
$$f_p \in \mathbb{N}_0 \ \forall \ p \in \mathcal{P}$$

We give all possible path flow solutions with corresponding costs in Table 2.1. In this network the best solution for order 1 is given in the first line of Table 2.1. The best solution for order 2 is given in the last line of Table 2.1.
To state how good a solution is for an order, we use the following ratio:

$$r_w = \frac{\sum_{p\in\mathcal{P}_w} C_w^p}{k_w} \ \forall \ w \in \{1, 2\}$$

where k_w is the total cost of the cheapest paths for order w (in this example the values of k_w are denoted in green in the columns C_1 and C_2 in Table 2.1). The SO solution (which is equal to the UE solution, according to the definitions used) is the solution of the last row in Table 2.1. In this network the worst ratio for order 2 can be $r_2 = \dfrac{640}{350} = 1.829$, which is not as bad as the ratio for order 1 in the SO solution (which is $r_1 = 6$). When looking purely at the ratios, it may be more fair to choose the situation in which both orders are disadvantaged equally. This is the case when the difference between the ratios is minimized: $\min\ |r_1 - r_2|$. This is the case in solution

$$f_{AB} = 1,\ f_{ACB} = 9,\ f_{AC} = 1,\ f_{ABC} = 19$$

with order cost for order 1 being $C_1 = 30 + 45 = 75$ and for order 2 being $C_2 = 3 + 608 = 611$. The ratios in this case are $r_1 = \frac{75}{50} = 1.5$ and $r_2 = \frac{611}{350} = 1.745714$, and in the last column of Table 2.1 we see the minimum ratio difference denoted in cyan.
It is also possible to look at the minimum total ratio, which in this case is the first line in Table 2.1, with the total ratio denoted in cyan.

Table 2.1: Solutions for Figure 2.6 with $d_1 = 10,\ d_2 = 20$.

f_{AB}	f_{ACB}	f_{AC}	f_{ABC}	$c_1^{AB} f_{AB}$	$c_1^{ACB} f_{ACB}$	$c_2^{AC} f_{AC}$	$c_2^{ABC} f_{ABC}$	C_1	C_2	r_1	r_2	$r_1 + r_2$	$\lvert r_1 - r_2 \rvert$
0	10	0	20	-	50	-	640	50	640	1	1.828571	2.83	0.83
1	9	1	19	30	45	3	608	75	611	1.5	1.745714	3.25	0.25
2	8	2	18	60	40	6	576	100	582	2	1.662857	3.66	0.34
3	7	3	17	90	35	9	544	125	553	2.5	1.58	4.08	0.92
4	6	4	16	120	30	12	512	150	524	3	1.497143	4.50	1.50
5	5	5	15	150	25	15	480	175	495	3.5	1.414286	4.91	2.09
6	4	6	14	180	20	18	448	200	466	4	1.331429	5.33	2.67
7	3	7	13	210	15	21	416	225	437	4.5	1.248571	5.75	3.25
8	2	8	12	240	10	24	384	250	408	5	1.165714	6.17	3.83
9	1	9	11	270	5	27	352	275	379	5.5	1.082857	6.58	4.42
10	0	10	10	300	-	30	320	300	350	6	1	7.00	5.00

The disadvantage of the last two solutions we denoted (the ones with minimum difference in ratio and minimum ratio sum), is the fact that the total costs are higher than the SO solution (with $f_{AB} = 10$, $f_{AC} = 10$, and $f_{ABC} = 10$). But there is a possibility to maintain the SO solution value, while minimizing the ratios, namely with toll pricing. We will discuss this in Chapters 4, 5, and 6. $\triangle$

2.1.4. Unique User Equilibrium path flows

Often there exists multiple User Equilibrium solutions concerning the path flow, if a solution exists. The arc flows $x_a\ \forall\ a \in \mathcal{A}$ are unique when cost functions of all arcs are strictly increasing (which is proven by Yang and Huang [30]). The existence of multiple path flow solutions can cause problems, because in this case different solvers can give different solutions. And when using the same solver, a small change of the network can cause a huge change in the path solution [7].
Can we force uniqueness? One option to do so is: For a given (unique) arc flow part of a UE one can compute a feasible route flow $\boldsymbol{f}$ that minimizes an objective G with fixed demand and arc flow:

$$\min\ G(\boldsymbol{f}) \text{ s.t. } \boldsymbol{f} \in \mathcal{F}.$$

The solution $\boldsymbol{f}$ is unique when G is strictly convex on $\mathcal{F}$ [17].

In this thesis we will assume the arc flows are not unique: there can be multiple solutions with equal arc flows, but with different path flows. We do not demand a unique User Equilibrium solution in the thesis.

2.1.5. Lagrangian Dual

We will use the Lagrangian Dual to show the correctness of Wardrops definition of a User Equilibrium and the commonly used objective function for those problems $\left(\int_0^{x_a} c_a(\tau)d\tau\right)$ in Subsection 2.1.6. We will first explain the Lagrangian Dual in this section.

The common used User Equilibrium problem in traffic assignment problems consists of an objective function with several equality and inequality constraints. In this section we consider a general notation of the IP:

$$\begin{aligned} z = \min_{\boldsymbol{x}} \; & g(\boldsymbol{x}) \\ \text{s.t.} \;\; & A\boldsymbol{x} \leq \boldsymbol{b} \\ & D\boldsymbol{x} = \boldsymbol{d} \\ & \boldsymbol{x} \in X \subseteq \mathbb{N}_0^n \end{aligned} \tag{IP}$$

We can compare this general form with the User Equilibrium IP in traffic assignment problems: take as objective function $g(\boldsymbol{x}) = \int_0^{x_a} c_a(\tau)d\tau$, with $A\boldsymbol{x} \leq \boldsymbol{b}$ the capacity constraints on the arcs and $D\boldsymbol{x} = \boldsymbol{d}$ the demand constraints.

We will now create a relaxation of the IP: For any values of $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_m) \in \mathbb{R}_+^m$, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_k) \in \mathbb{R}^k$ we define the problem

$$\begin{aligned} z(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min \; & g(\boldsymbol{x}) + \boldsymbol{\lambda}(A\boldsymbol{x} - b) + \boldsymbol{\mu}(D\boldsymbol{x} - \boldsymbol{d}) \\ \text{s.t.} \;\; & \boldsymbol{x} \in X \end{aligned} \tag{IP($\boldsymbol{\lambda}, \boldsymbol{\mu}$)}$$

Where (IP($\boldsymbol{\lambda}, \boldsymbol{\mu}$)) is a relaxation of (IP) for all $\boldsymbol{\lambda} \geq 0$, $\boldsymbol{\mu} \in \mathbb{R}^k$ [28, Proposition 10.1]. This holds because the feasible region of (IP($\boldsymbol{\lambda}, \boldsymbol{\mu}$)) is at least as large as the feasible region of (IP), because

$$\{\boldsymbol{x} \mid A\boldsymbol{x} \leq \boldsymbol{b},\; D\boldsymbol{x} = \boldsymbol{d},\; \boldsymbol{x} \in X\} \subseteq X.$$

Also the objective value is at least as small in (IP($\boldsymbol{\lambda}, \boldsymbol{\mu}$)) as in (IP) for all feasible solutions in (IP), because as $\boldsymbol{\lambda} \geq 0$ and $A\boldsymbol{x} \leq \boldsymbol{b}$ for all $\boldsymbol{x} \in X$, then $g(\boldsymbol{x}) + \boldsymbol{\lambda}(A\boldsymbol{x} - \boldsymbol{b}) \leq g(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$.
$\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are called the Lagrange vectors. Problem (IP($\boldsymbol{\lambda}, \boldsymbol{\mu}$)) is called a Lagrangian relaxation of (IP) with parameters $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$. With the dual we can find a lower bound on our minimization problem.

The Lagrangian Dual problem is denoted as:

$$\begin{aligned} w_{LD} &= \max\{z(\boldsymbol{\lambda}, \boldsymbol{\mu}) \mid \boldsymbol{\lambda} \geq 0\} \\ &= \max_{\boldsymbol{\lambda} \geq 0, \boldsymbol{\mu}} z(\boldsymbol{\lambda}, \boldsymbol{\mu}) \end{aligned} \tag{LD}$$

The Lagrangian Dual can give us a lower bound on our minimization problem (IP) and sometimes even leads to an optimal solution of (IP) [28].

Proposition 1. *If $\boldsymbol{\lambda} \geq 0$,*

1. *$\boldsymbol{x}(\boldsymbol{\lambda})$ is an optimal solution of IP($\boldsymbol{\lambda}$), and*
2. *$A\boldsymbol{x}(\boldsymbol{\lambda}) \leq \boldsymbol{b}$, and*
3. *$(A\boldsymbol{x}(\boldsymbol{\lambda}))_i = b_i$ whenever $\lambda_i > 0$ (complementarity),*

then $\boldsymbol{x}(\boldsymbol{\lambda})$ is optimal in IP. [28, Proposition 10.2]

For equality constraints ($\boldsymbol{\mu}$ in our problem): Condition (3) is automatically satisfied, so an optimal solution to IP($\boldsymbol{\mu}$) is optimal for IP if it is feasible in IP.

The question for now is, is the Lagrangian Dual useful to solve our User Equilibrium Problem? Therefore we need to know how good the bound is, which is obtained by solving the Lagrangian Dual, and how to actually solve it.
The strength of a bound can be expressed by the following theorem:

Theorem 1. $w_{LD} = \min\{g(\boldsymbol{x}) \mid A\boldsymbol{x} \leq \boldsymbol{b},\ D\boldsymbol{x} = \boldsymbol{d},\ \boldsymbol{x} \in conv(X)\}$. *[28, theorem 10.3]*

Proof.

$$\begin{aligned}
w_{LD} &= \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\mu}} z\,(\boldsymbol{\lambda},\boldsymbol{\mu}) \\
&= \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\mu}} \left\{\min_{\boldsymbol{x}\in X} [g(\boldsymbol{x}) + \boldsymbol{\lambda}\,(A\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{\mu}\,(D\boldsymbol{x} - \boldsymbol{d})]\right\} \\
&= \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\mu}} \left\{\min_{t=1,\dots,T} [g(\boldsymbol{x}_t) + \boldsymbol{\lambda}\,(A\boldsymbol{x}_t - \boldsymbol{b}) + \boldsymbol{\mu}\,(D\boldsymbol{x}_t - \boldsymbol{d})]\right\} \\
&= \max \eta \\
&\quad \text{s.t. } \eta \leq g(\boldsymbol{x}_t) + \boldsymbol{\lambda}\,(A\boldsymbol{x}_t - \boldsymbol{b}) + \boldsymbol{\mu}\,(D\boldsymbol{x}_t - \boldsymbol{d}) \quad \text{for all } t \\
&\qquad\quad \boldsymbol{\lambda} \in \mathbb{R}_+^T,\ \boldsymbol{\mu} \in \mathbb{R}^T,\ \eta \in \mathbb{R}
\end{aligned}$$

where the set X consist of a finite number of points $\{x_1, \dots, \boldsymbol{x}_T\}$, and the chosen $\boldsymbol{x}_t$ denotes the optimal solution such that the expression $z(\boldsymbol{\lambda},\boldsymbol{\mu})$ is minimized. The new variable η is introduced to represent a lower bound on $z(\boldsymbol{\lambda},\boldsymbol{\mu})$.
Taking the dual of the latter (linear) problem gives:

$$\begin{aligned}
w_{LD} &= \max_{u_t}\left\{\min_{\eta\in\mathbb{R}} -\eta + \sum_{t=1}^{T} u_t\,(\eta - g(\boldsymbol{x}_t) - \boldsymbol{\lambda}\,(A\boldsymbol{x}_t - \boldsymbol{b}) - \boldsymbol{\mu}\,(D\boldsymbol{x}_t - \boldsymbol{d}))\right\} \\
&= \max_{u_t}\left\{\min_{\eta\in\mathbb{R}} \eta\left(\sum_{t=1}^{T}(u_t) - 1\right) + \sum_{t=1}^{T} u_t\,(-g(\boldsymbol{x}_t) - \boldsymbol{\lambda}\,(A\boldsymbol{x}_t - \boldsymbol{b}) - \boldsymbol{\mu}\,(D\boldsymbol{x}_t - \boldsymbol{d}))\right\} \\
&= \min_{u_t}\left\{\sum_{t=1}^{T} u_t\,(g(\boldsymbol{x}_t) + \boldsymbol{\lambda}\,(A\boldsymbol{x}_t - \boldsymbol{b}) + \boldsymbol{\mu}\,(D\boldsymbol{x}_t - \boldsymbol{d}))\right\} \\
&\quad \text{s.t. } \boldsymbol{\lambda} \in \mathbb{R}_+^T,\ \boldsymbol{\mu} \in \mathbb{R}^T,\ \boldsymbol{u} \in \mathbb{R}^T.
\end{aligned}$$

This equivalent to

$$\begin{aligned}
w_{LD} = \min\ & \sum_{t=1}^{T} u_t g(\boldsymbol{x}_t) \\
\text{s.t. } & \sum_{t=1}^{T} u_t(A\boldsymbol{x}_t - \boldsymbol{b}) \leq 0 \\
& \sum_{t=1}^{T} u_t(D\boldsymbol{x}_t - \boldsymbol{d}) = 0 \\
& \sum_{t=1}^{T} u_t = 1 \\
& \boldsymbol{u} \in \mathbb{R}_+^T
\end{aligned}$$

Setting $\boldsymbol{x} = \sum_{t=1}^{T} u_t \boldsymbol{x}_t$, with $\sum_{t=1}^{T} u_t = 1$, $\boldsymbol{u} \in \mathbb{R}_+^T$ we get:

$$\begin{aligned}
w_{LD} = \min\ & g(\boldsymbol{x}) \\
\text{s.t. } & A\boldsymbol{x} \leq \boldsymbol{b} \\
& D\boldsymbol{x} = \boldsymbol{d} \\
& \boldsymbol{x} \in \text{conv}(X)
\end{aligned}$$

□

To solve the formulation in Theorem 1 we need a constraint generation (or cutting plane) approach, because of the large amount of constraints.
Wolsey also gives a Sub-gradient algorithm for the Lagrangian Dual, where you start with an initial Lagrangian multiplier which you adjust every step.

Algorithm 1 Sub-gradient algorithm for the Lagrangian Dual [28]

Initialization. $\boldsymbol{u} = \boldsymbol{u}^0$.
Iteration k. $\boldsymbol{u} = \boldsymbol{u}^k$.
while $\boldsymbol{u}^k \neq \boldsymbol{u}^{k+1}$
do Solve the Lagrangian Problem IP(u^k) with optimal solution $\boldsymbol{x}(\boldsymbol{u}^k)$.
$\boldsymbol{u}^{k+1} = \max\{\boldsymbol{u}^k - \mu_k(d - D\boldsymbol{x}(\boldsymbol{u}^k)), 0\}$
$k \leftarrow k + 1$.
end while

$\boldsymbol{d} - D\boldsymbol{x}(\boldsymbol{u}^k)$ is a sub-gradient of $z(\boldsymbol{u})$ at $\boldsymbol{u}^k$. The difficulty of this algorithm is in choosing the step lengths $\{\mu_k\}_{k=1}^{\infty}$:

Theorem 2. *[28, theorem 10.4]*

(a) *If $\sum_k \mu_k \to \infty$, and $\mu_k \to 0$ as $k \to \infty$, then $z(\boldsymbol{u}^k) \to w_{LD}$ is the optimal value of LD.*

(b) *If $\mu_k = \mu_0 \rho^k$ for some parameter $\rho < 1$, then $z(\boldsymbol{u}^k) \to w_{LD}$ if μ_0 and ρ are sufficiently large.*

(c) *If $\overline{w} \geq w_{LD}$ and $\mu_k = \epsilon_k[z(\boldsymbol{u}^k) - \overline{w}]/||\boldsymbol{d} - D\boldsymbol{x}(\boldsymbol{u}^k)||^2$ with $0 < \epsilon_k < 2$, then $z(\boldsymbol{u}^k) \to \overline{w}$, or the algorithm finds $\boldsymbol{u}^k$ with $\overline{w} \geq z(\boldsymbol{u}^k) \geq w_{LD}$ for some finite k.*

2.1.6. Lagrangian duality in User Equilibrium problems

We will now use the Lagrangian Dual to show the correctness of Wardrops definition of a User Equilibrium and the commonly used objective function for those problems: $\int_0^{x_a} c_a(\tau)d\tau$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a directed graph and let $\mathcal{P}$ be the set of paths p in the graph $\mathcal{G}$. Denote $\mathcal{P}_w$ as the set of paths for OD-pair $w \in \mathcal{W}$, and $\mathcal{P} = \bigcup_{w \in \mathcal{W}} \mathcal{P}_w$. The User Equilibrium problem is given by

$$\min \sum_{a \in \mathcal{A}} \int_0^{x_a} c_a(\tau)d\tau \tag{2.22}$$

$$\text{s.t.} \sum_{p \in P_w} f_p = d_w \;\forall\; w \in W \tag{2.23}$$

$$x_a = \sum_{p \in P} \delta_{ap} f_p \;\forall\; a \in \mathcal{A} \tag{2.24}$$

$$f_p \in \mathbb{N}_0 \;\forall\; p \in P \tag{2.25}$$

with $\delta_{ap} = \begin{cases} 1 & \text{if } a \text{ is contained in } p, \\ 0 & \text{otherwise.} \end{cases}$

When we interpret the UE as a situation in which each order travels via its cheapest paths [1], we can state

$$C_p(\boldsymbol{f}) = \min_{q \in \mathcal{P}_w} C_q(\boldsymbol{f}) \;\forall\; p \in \mathcal{P}_w \text{ s.t. } \; f_p > 0,\; f_p \in \mathbb{N} \;\forall\; w \in \mathcal{W}.$$

Beckmann et al. [6] proved that such a flow always exists by considering the following min-cost multi-commodity flow problem with separable objective function (see Subsection 2.1.2)

$$\min \sum_{a \in \mathcal{A}} \int_0^{x_a} h_a(\tau)d\tau.$$

In the UE problem, which can have a nonlinear objective, (2.22) is the objective function, with Constraints 2.23, which states that for every OD-pair w, the demand is provided by the paths corresponding to this OD-pair and Constraint 2.24 states that the flow on arc a is the sum of all path flows that make use of this arc.
A Lagrangian Dual of this problem is:

$$\begin{aligned} w_{LD} &= \max_{\lambda_w \in \mathbb{R}} \min_{f \geq 0} Z_{\boldsymbol{\lambda}} \\ &= \max_{\lambda_w \in \mathbb{R}} \min_{f_p \in \mathbb{N}_0} \sum_{a \in A} \int_0^{x_a} c_a(\tau) d\tau + \sum_w \lambda_w \left(d_w - \sum_{p \in P_w} f_p \right) \\ \text{s.t. } x_a &= \sum_{p \in P} \delta_{ap} f_p \ \forall \ a \in \mathcal{A} \\ f_p &\in \mathbb{N}_0 \ \forall \ p \in \mathcal{P} \end{aligned}$$

Theorem 3 (Complementary Slackness). *[2] Let f_p, $p \in P$ and λ_w, $w \in W$ be feasible solutions to the primal and dual problem, respectively. They are optimal solutions for the two respective problems if and only if*

$$\lambda_w \left(\sum_{p \in P_w} f_p - d_w \right) = 0 \quad \forall \ w \in \mathcal{W} \tag{2.26}$$

and

$$f_p \left(c_a \left(\sum_{p \in P} \delta_{ap} f_p \right) - \lambda_w \right) = 0 \quad \forall \ p \in \mathcal{P} \tag{2.27}$$

In Equality 2.26 we see the that the part we added to the objective function must be equal to 0. With Equality 2.27 it can easily be verified that a path p can only be used when the costs of using this path is equal to the minimum cost for OD-pair w. This minimum cost is thus denoted by λ_w. Because of the fact that $f_p \in \mathbb{N}_0 \ \forall \ p \in \mathcal{P}$ we can conclude that in the Dual Problem, the constraints

$$c_a \left(\sum_{p \in \mathcal{P}} \delta_{ap} f_p \right) - \lambda_w \geq 0 \ \forall \ p \in P$$

have to hold. This also is valid because the cost of a path p can never be smaller than the minimum path cost λ_w.

2.2. Tolls systems in the literature

Instead of solving the UE as we done in the section before, we can solve UE problems as well by using tolls. In this section we will give an outline of toll in the literature, and in Chapters 4, 5, and 6 we will show our own method of toll pricing.

Hearn and Ramana [13] make use of a toll pricing system by adding a toll term to the cost function for each arc. They define the system of feasible flows:

$$x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \ \forall \ a \in \mathcal{A} \tag{2.28}$$

$$\sum_{p \in \mathcal{P}_w} f_p = d_w \ \forall \ w \in \mathcal{W} \tag{2.29}$$

$$f_p \in \mathbb{N}_0 \ \forall \ p \in \mathcal{P} \tag{2.30}$$

$$F = \left\{ (\boldsymbol{x}, \boldsymbol{f}) \ | \ x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \ \forall \ a \in \mathcal{A}, \ \sum_{p \in \mathcal{P}_w} f_p = d_w \ \forall \ w \in \mathcal{W}, \ f_p \in \mathbb{N}_0 \ \forall \ p \in \mathcal{P} \right\}$$

$$V = \{\boldsymbol{x} \ | \ \text{ there exists } \boldsymbol{f} \text{ s.t. } (\boldsymbol{x}, \boldsymbol{f}) \in \ F\}$$

where Constraint 2.28 denotes the total arc flow, given all path flows using that arc, Constraint 2.29 the demand constraint, Constraint 2.30 assures all flows are nonnegative, F is the set of all feasible flows and V the set of feasible arc flows.

The cost function on arc a is $c_a(x_a)$ and the amount of toll on arc a is denoted by β_a, which can be both positive and negative. Then the adjusted arc costs as perceived by a customer are given by

$$\boldsymbol{c}_\beta(\boldsymbol{x}) = \boldsymbol{c}(\boldsymbol{x}) + \boldsymbol{\beta}.$$

with arc flow vector $\boldsymbol{x}$, cost vector $\boldsymbol{c}(\boldsymbol{x})$, toll vector $\boldsymbol{\beta}$ and tolled cost vector $\boldsymbol{c}_\beta(\boldsymbol{x})$. A tolled equilibrium then is

$$\boldsymbol{c}_\beta(\overline{\boldsymbol{x}})^T(\boldsymbol{x} - \overline{\boldsymbol{x}}) \geq 0 \ \forall \ \boldsymbol{x} \in V$$

with $\overline{\boldsymbol{x}}$ the UE solution vector.
Define the set of tolled equilibrium solutions:

$$U_\beta^* := \{\overline{\boldsymbol{x}} \ | \ (\boldsymbol{c}(\overline{\boldsymbol{x}}) + \boldsymbol{\beta})^T(\boldsymbol{x} - \overline{\boldsymbol{x}}) \geq 0 \ \forall \ \boldsymbol{x} \in V\}.$$

The SO solution set is given by

$$S^* := \text{argmin}\{\boldsymbol{c}(\boldsymbol{x})^T \boldsymbol{x} \ | \ \boldsymbol{x} \in V\}.$$

They require

$$\emptyset \neq U_\beta^* \subseteq S^*, \tag{2.31}$$

because we want the resulting UE has a solution and every equilibrium solution is SO. If a toll vector $\boldsymbol{\beta}$ suffices (2.31), it is a valid toll vector.
The set of all these toll vectors is

$$\mathcal{T} := \{\boldsymbol{\beta} \ | \ \emptyset \neq U_\beta^* \subseteq S^*\}.$$

Also define

$$W(\overline{\boldsymbol{x}}) = \{\boldsymbol{\beta} \ | \ \overline{\boldsymbol{x}} \in U_\beta^*\}, \text{ with } \overline{\boldsymbol{x}} \in V$$

and we can say

$$W(\overline{\boldsymbol{x}}) = N(\overline{\boldsymbol{x}}; V) - \boldsymbol{c}(\overline{\boldsymbol{x}}),$$

where $N(\overline{\boldsymbol{x}}, V) = \{\boldsymbol{u} \mid \boldsymbol{u}^t(\boldsymbol{x} - \overline{\boldsymbol{x}}) \geq 0 \; \forall \; \boldsymbol{x} \in V\}$. From Hearn and Ramana [13] also follows that the toll vector set can be denoted as

$$\mathcal{T} = \bigcup_{\boldsymbol{x}^* \in S^*} W(\boldsymbol{x}^*)$$

Hearn and Ramana also give a problem formulation for Mathematical Programs with Equilibrium Constraints with tolls:

$$\begin{aligned}
\min \; & \sum_{a \in \mathcal{A}} c_a(\overline{x}_a)\overline{x}_a \\
\text{s.t.} \; & \overline{\boldsymbol{x}} \in V \\
& (\boldsymbol{c}(\overline{\boldsymbol{x}}) + \boldsymbol{\beta})^T(\boldsymbol{x} - \overline{\boldsymbol{x}}) \geq 0 \; \forall \; \boldsymbol{x} \in V. \\
& x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \; \forall \; a \in \mathcal{A} \\
& \sum_{p \in \mathcal{P}_w} f_p = d_w \; \forall \; w \in \mathcal{W} \\
& f_p \in \mathbb{N}_0 \; \forall \; p \in \mathcal{P}
\end{aligned} \tag{MPEC}$$

This problem has both $\boldsymbol{\beta}$ and $\overline{\boldsymbol{x}}$ as its variables. Our path based toll model is based on the MPEC of Hearn and Ramana.
They also describe the Robinhood formulation, in which the sum of all tolls must be zero, so that there is no profit for the system: all money that is payed by certain customers in the system will arrive at other customers inside the system.
In this case they calculate the toll after a System Optimal solution is found. Then they state the following ILP:

$$\begin{aligned}
\min \; & \sum_{a \in \mathcal{A}} \beta_a x_a \\
\text{s.t.} \; & \sum_{a \in \mathcal{A}} \beta_a x_a = 0
\end{aligned} \tag{2.32}$$

According to Florian and Hearn [10], the application of those types of toll is hard to implement on traffic networks regarding variable travel times, although the selective use of negative tolls to influence route choice of users might have some appeal. In this thesis we will show a way of applying both negative and positive tolls.

Larsson et. al. [15] give an algorithm to calculate tolls to links in such a way that the tolls force individual drivers to choose routes such that the total costs and environmental solution are minimized (so finding an SO solution) while drivers still search for their own optimal route in ways of travel time and costs.
The question is how to divide tolls over the network. As stated by Small [23]: “the people who benefit from congestion relief and revenue uses do not necessarily coincide with those who pay the fees or who suffer inconvenience in order to avoid them.” We state that fairness is reached when customers who are not causing any bottleneck, should not pay tolls which solve that certain bottleneck, therefore we divide the customers in different sets, as we will show in Chapter 5.

Yang and Han [12] investigate the use of tolls with the help of the price of anarchy. Yang and Zhang [32] constructed an anonymous link toll system to add traveler-dependent tolls. They concluded that there exist nonnegative links tolls identical to all users to decentralize the Wardropian System Optimum as a UE-CN (Cournot-Nash) mixed equilibrium, and the valid toll set is made up of a convex set of linear equalities and inequalities. Yang and Han [12] and Yang and Zhang [32] use nonnegative tolls. Our approach is to use tolls that sum up to zero. Yang and Huang [31] state that Value Of Time (VOT) is a very important concept in transportation system modeling. The VOT of an order is a constant which denotes the importance of that customer. We will assume the VOT is equal for all customers in the network, so all time based costs have the same urgency for each customer.

Didi-Biha, Marcotte and Savard [9] also use nonnegative tolls. Their goal is to maximize the toll revenue for the highway authority while the users of the network want to minimize their traveling costs. They introduce their bi-level programming Toll Optimization Problem, both arc, arc-path and path based.

Yang [30] proved the existence of a Pareto refunding scheme that returns the congestion pricing revenues to all users to make everyone better off. This Pareto refunding scheme refunds class-specific and OD-specific toll revenue equally to all users in the same OD pair in the same user class.

3

User Equilibria and Tolls in Space Time Networks

In this chapter, we will discuss the User Equilibrium problem on Space Time Networks. In Subsection 1.2.2 we introduced the definition of a UE by Wardrop, in this chapter we will formulate our demands for the definition of a User Equilibrium. In Section 3.1 we will show how we can solve the UE problem on the Space Time Network. In Section 3.2 we introduce the use of tolls on the STN and in Section 3.3 we introduce the definitions of User Equilibria for different types of tolls.

3.1. User Equilibrium on Space Time Networks

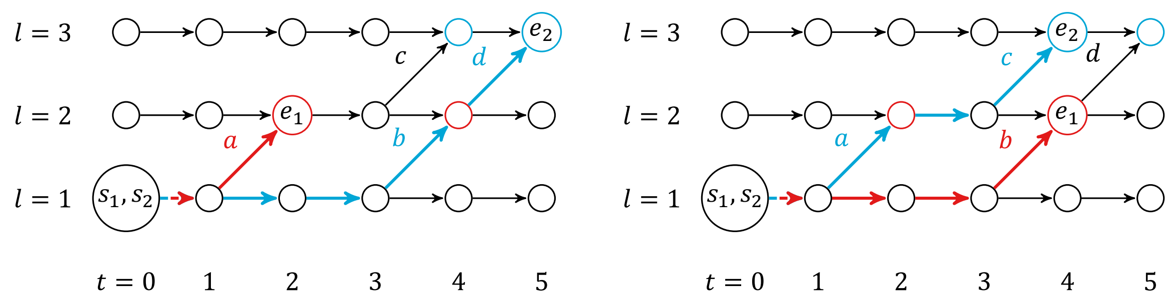
To define the User Equilibrium on Space Time Networks, we need to know what conditions the solutions need to meet. To find those conditions, we will use Example 5 and Example 6. But first we introduce some notation: recall that we denote the path cost by C^p or C^p_w. The total order cost is denoted by C_w for all w, with

$$C_w = \sum_{p \in \mathcal{P}_w} C^p f_p = \sum_{p \in \mathcal{P}_w} C^p_w f_p.$$

We denote the cheapest path cost for an order by k_w. We calculate this cheapest path cost by assigning containers of order w to paths assuming this order is the only one traveling in the STN.
We will show two examples to explain how the STN can be used and how we can compare different solutions with each other.

Example 5. In this example there are three locations, $\mathcal{V} = \{1, 2, 3\}$, and five time steps. We have two connections between location $l = 1$ and $l = 2$ and two between $l = 2$ and $l = 3$. Those arcs all have capacity $m_a = 1$, and $m_a = \infty$ for waiting arcs. We have two orders, order 1 and 2 both start at location 1, order 1 has to go to $l = 2$ and order 2 to $l = 3$. Every node column shows a time step and each arc has cost $c_a = 1$. The two possible solutions are given in Figure 3.1, with s_w denoting the starting point and e_w denoting the end point for order w.

In Figure 3.1a we see the SO solution, that is the solution where the total costs are minimized. Here order 1 is delivered first with cost $C^a_1 = 2$ and therefore order 2 can only take path bd with cost $C^{bd}_2 = 5$. In Figure 3.1b a solution is given where both orders pay cost 4, that is path b for order 1, and path ac for order 2.
We can see that each order has its own preferable solution, that is the one in which they can travel via arc a, which is in the cheapest path for both orders.



(a) The SO solution.

(b) The alternative solution.

Figure 3.1: STN with two orders, with $m_a = 1$ for all arcs between two different locations, $m_a = \infty$ otherwise.

The question now is, can we call one of those solutions a User Equilibrium? Assuming customers choose their own routes, the solution depends on which customer gets to choose first. According to Wardrops definition (see Section 2.1) a User Equilibrium is obtained if the costs for all customers are equal (so the situation in Figure 3.1b), but we have to keep in mind that this equality only holds if the customers are having the same OD-pair. According to the objective function for UE in traffic assignment problems, the UE solution is equal to the SO solution, due to the constant arc costs (as we also saw in (2.12)):

$$\min\ z_{UE} = \min \sum_{a \in \mathcal{A}} \int_0^{x_a} c_a(\tau)d\tau = \min \sum_{a \in \mathcal{A}} c_a x_a = \min \sum_{p \in \mathcal{P}} C^p f_p = z_{SO}.$$

Thus, for the definitions as used in the literature, the solution given in Figure 3.1a is the best User Equilibrium, because in this solution z_{UE} is minimized. But we can easily verify order 2 is not satisfied with this solution.

We can also look at the ratio between the cheapest path and the alternative path per customer in this example, to see how worse it will be for an order to travel via its alternative path. A ratio of 1 means that the order can travel via its cheapest path, and thus is satisfied.
For order 1, the cheapest path cost is $k_1 = C_1^a = 2$, the alternative path cost is $C_1^b = 4$. Then we obtain the ratio $r_1 = \frac{C_1}{k_1} = 2$. For order 2, the cheapest path cost is $k_2 = C_2^{ac} = 4$, the alternative path cost is $C_2^{bd} = 5$. Then we obtain the ratio $r_2 = \frac{C_2}{k_2} = \frac{5}{4}$.

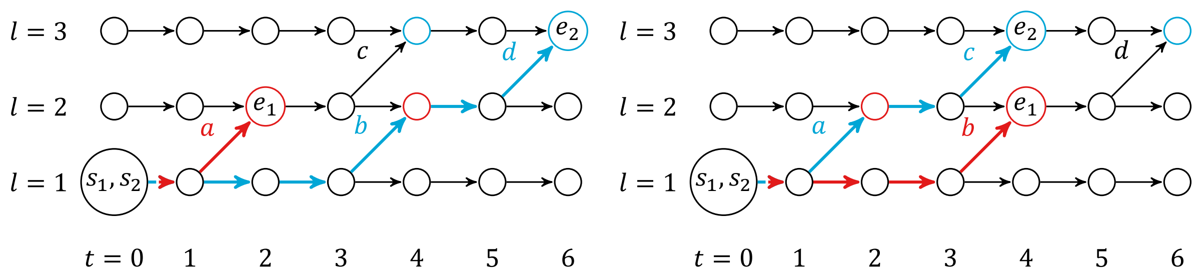
We see that both order 1 and order 2 want to travel via arc a in order to travel via their cheapest path. If order 1 is forced to take his most expensive path b in order to let order 2 travel via its cheapest path ac, it is more expensive for order 1 to travel via its second-best path than when order 2 is forced to travel via its second-best path (because $r_1 = 2 > r_2 = \frac{5}{4}$). So we could conclude that letting order 2 traveling via path bd and letting order 1 take its cheapest path, is the User Equilibrium (thus in this case the SO solution, denoted in Figure 3.1a), because in this solution all customers are harmed as little as possible. △

We can conclude from this example, that we want to minimize the harm of traveling via alternative paths, which are all paths except for the cheapest path. We will use the ratio to denote the harm of a customer in the current compared to its cheapest path:

$$r_w = \frac{C_w}{k_w}\ \forall\ w \in \mathcal{W}.$$

So the ratio is the cost payed by an order divided by the cheapest path costs of that order. We will endorse the benefit of the minimization of harm by the following example:

Example 6. We adjusted Example 5: we added one extra time step and moved arc d one time step. The two possible solutions are given in Figure 3.2, with s_w denoting the starting point and e_w denoting the end point for order w.



(a) SO solution 1 (b) SO solution 2

Figure 3.2: STN with two orders, with $m_a = 1$ for all arcs between two different locations, $m_a = \infty$ otherwise.

In this case there are two SO solutions, because both solutions have the same objective cost: $C_1 + C_2 = 8$.
We compare the two solutions by calculating the ratio of the current solution divided by the cheapest path for each individual order (again arc a is in the cheapest paths for both orders):
Solution in Figure 3.2a:
Order 1: $k_1 = C_1^a = 2$, chosen path cost $C_1^a = 2$: Ratio $r_1 = \frac{C_1}{k_1} = \frac{2}{2} = 1$.
Order 2: $k_2 = C_2^{ac} = 4$, chosen path cost $C_2^{bd} = 6$: Ratio $r_2 = \frac{C_2}{k_2} = \frac{6}{4} = \frac{3}{2}$.

Solution in Figure 3.2b:
Order 1: $k_1 = C_1^a = 2$, chosen path cost $C_1^b = 4$: Ratio $r_3 = \frac{C_3}{k_3} = \frac{4}{2} = 2$.
Order 2: $k_2 = C_2^{ac} = 4$, chosen path cost $C_2^{ac} = 4$: Ratio $r_4 = \frac{C_4}{k_4} = \frac{4}{4} = 1$.
The higher the ratio is, the more the order is disadvantaged. So in order to have satisfied customers, we want all ratios to be as low as possible. When looking at the example, then the solution as given in Figure 3.2a is the best User Equilibrium, so order 1 traveling via path a and order 2 via bd. △

3.1.1. Using ratios

As shown in the section above, finding a solution in which all orders are satisfied, is not always possible. A way of defining a User Equilibrium where all orders are harmed as little as possible, is looking at the ratios of the used path cost compared to the cheapest path cost for that order. We can state this in an ILP with objective function

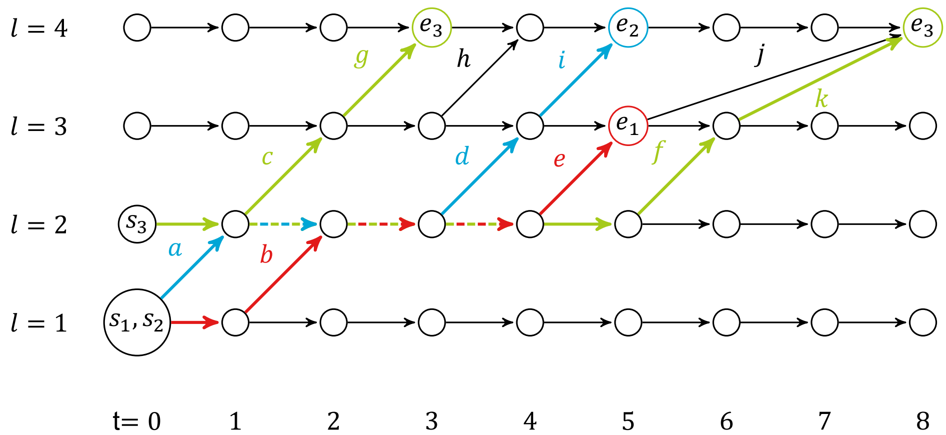
$$\min \sum_{w \in \mathcal{W}} r_w = \min \sum_{w \in \mathcal{W}} \frac{C_w}{k_w} = \min \sum_{w \in \mathcal{W}} \left(\frac{1}{k_w} \sum_{p \in \mathcal{P}_w} C_w^p f_p \right).$$

We use the following ILP:

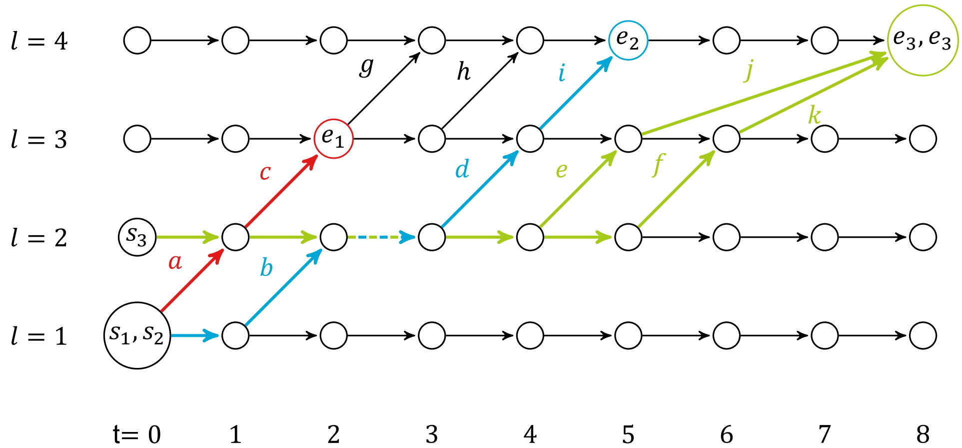
$$\begin{aligned}
\min \quad & \sum_{w \in \mathcal{W}} \left(\frac{1}{k_w} \sum_{p \in \mathcal{P}_w} C_w^p f_p \right) \\
\text{s.t.} \quad & \sum_{p \in P_w} f_p = d_w \ \forall\ w \in W \\
& x_a = \sum_{p \in P} \delta_{ap} f_p \ \forall\ a \in \mathcal{A} \\
& f_p \in \mathbb{N}_0 \ \forall\ p \in P
\end{aligned}$$

We refer to this objective function as the Weighted Ratio objective. In the following example we will show the influence of using this objective function.

Example 7. In this example we have three orders, order 1 traveling from location $l = 1$ to $l = 3$, order 2 from $l = 1$ to $l = 4$ and order 3 from $l = 2$ to $l = 4$. The demands are $d_1 = d_2 = 1$ and $d_3 = 2$. The UE solution is given in Figure 3.3a and the Weighted Ratio solution is given in Figure 3.3b, the values of the solutions can also be read in Table 3.1.



(a) SO solution.



(b) Weighted Ratio solution.

Figure 3.3: STN with three orders, with $m_a = 1$ for all arcs between two different locations, $m_a = \infty$ otherwise.

Table 3.1: Solution of STN in Figure 3.3a

	Order 1	Order 2	Order 3	Total Cost	Total Ratio
cheapest paths cost k_w	2	3	8		
Solution Weighted Ratio	2	5	16	23	
Solution SO	5	5	11	21	
Ratio of Weighted Ratio r_w	1	1.6667	2		4.66667
Ratio of SO r_w	2.5	1.6667	1.375		5.54167

We see that the total costs of the Weighted Ratio solution are higher than those of the SO solution. Note that the SO solution is not unique: the costs of the containers of orders 2 and 3 can be switched without influencing the total costs. The benefit of using the Weighted Ratio objective function, is the fact that the total ratio is minimized. It is less harmfull for order 3 to pay more than it is for order 1, compared to their cheapest path costs. △

The question now arises, what do we rather achieve? When looking for a User Equilibrium, do we want the total costs minimized (SO), or do we want to divide the extra costs fairly compared to cheapest path costs (Weighted Ratio)? The best solution would be a combination, in which we obtain an SO, and the extra costs made by having to travel via non-cheapest paths are divided in a fair way over the orders. In the next section we will introduce a method to find this combination, by adding tolls the the network, and we will give definitions of a User Equilibrium.

3.2. Toll In Space Time Networks

Our goal is to reach as well an SO solution regarding the total costs, as achieving a UE. By a SO solution we mean a solution in which the total costs are minimized. However, it is always the case that a UE has a total cost equal or higher to the total cost of the SO solution. To avoid this higher total cost in a UE, we introduce the use of tolls in three ways:

1. Tolls on orders: assign tolls to the orders after obtaining the SO solution, to create a UE solution in which costs are divided over the orders in a fair way.
2. Tolls on paths: assign tolls to paths after obtaining the SO solution and create tolled path costs on the STN, in which the paths of the SO solution have cheapest (tolled) paths costs and thus a UE solution is obtained.
3. Tolls on arcs: assign tolls to arcs after obtaining the SO solution and create tolled arc costs on the STN, in which the paths of the SO solution have cheapest (tolled) path costs and thus a UE solution is obtained.

The outline of the procedure is to calculate the SO solution in the STN, then add tolls (which can be both positive and negative) to arcs, paths or orders, in such a way that the total sum of all tolls on orders (in the order case), on used paths (in the path case) or on used arcs (in the arc case), is equal to zero. We will assume that all OD-pairs are individual customers.

In this thesis we will investigate the first two ways of adding tolls: on orders and on paths. We use the following notation for tolls: the toll on path is β_w^p and the toll on order is given by

$$\beta_w = \sum_{p \in \mathcal{P}_w} \beta_w^p f_p.$$

We define the tolled order costs:

$$C_{\beta w} = C_w + \beta_w$$

as the order cost including the tolls, and the tolled path costs:

$$C_{\beta w}^p = C_w^p + \beta_w^p$$

as the path cost including the tolls. Note that

$$C_{\beta w} = C_w + \beta_w = \sum_{p \in \mathcal{P}_w} \left(C_w^p + \beta_w^p\right) = \sum_{p \in \mathcal{P}_w} C_{\beta w}^p \ \forall\, w \in \mathcal{W}.$$

k_w is the cost of the cheapest paths for order w.

We can compare the tolls on orders to real life instances. We see in real life instances that adding toll on orders is as assigning “fair” costs to customers which use the same transport modalities (multiple containers on the same train or barge). We assume when assigning toll to orders, that the customer is familiar with the initial STN, but that the tolls will give final costs that satisfy a certain “fairness” criterion. We will discuss this in Chapter 5.

Adding toll on paths gives us a new cost network wherein the goal is that each customer will choose paths which are also assigned to him in the SO solution. We assume that the customers only know the new cost network (and thus do not know the initial costs of all paths).

We can say a UE is reached when the costs of the paths are defined such that in the case all capacity constraints are removed, still the same paths will be chosen by the customers. We will discuss this in Chapter 4.

To investigate the behavior of tolls, we will divide MCMCF problem on an STN into different cases:

- Case 1: demand $d_w = 1$ $\forall$ orders $w \in \mathcal{W}$, capacity $m_a = 1$ $\forall$ a between two different locations, $m_a = \infty$ otherwise, $a \in \mathcal{A}$,
- Case 2: demand $d_w \geq 1$ $\forall$ orders $w \in \mathcal{W}$, capacity $m_a = 1$ $\forall$ a between two different locations, $m_a = \infty$ otherwise, $a \in \mathcal{A}$,
- Case 3: demand $d_w = 1$ $\forall$ orders $w \in \mathcal{W}$, capacity $m_a \geq 1$ $\forall$ a between two different locations, $m_a = \infty$ otherwise, $a \in \mathcal{A}$,
- Case 4: demand $d_w \geq 1$ $\forall$ orders $w \in \mathcal{W}$, capacity $m_a \geq 1$ $\forall$ a between two different locations, $m_a = \infty$ otherwise, $a \in \mathcal{A}$.

When investigating adding tolls, we use the different cases described above. In all cases, for all waiting arcs, thus horizontal arcs in an STN, we define $m_a = \infty$, so there is no capacity constraint for waiting at a location.

3.3. User Equilibrium with tolls on a Space Time Network

We define a User Equilibrium per situation which we will investigate, so we have a definition for path based tolls and one for order based tolls:

Path based tolls: For each order there are multiple paths to travel by. Here we define User Equilibrium as follows:

Definition 4 (Path based User Equilibrium)**.** A User Equilibrium is reached when each customer can travel via their cheapest paths.

This is obviously not always possible when concerning only the initial networks, so we need the path tolls for this. We will assume that customers are not familiar with the path costs in the initial STN, they only have knowledge of the tolled path costs.

Order based tolls: When we observe the costs payed by each order, we can define a "fairness" criterion in order to satisfy customers. For each customer in the problem, we introduce the ratio between the order costs of the SO solution C_w and the order costs if there were no other orders in the network, so the cost of the cheapest paths for order $w \in \mathcal{W}$: k_w.
The ratio for all orders $w \in \mathcal{W}$ is $r_w = \frac{C_w}{k_w}$ and the ratio of tolled order costs are denoted by $r_{\beta w} = \frac{C_{\beta w}}{k_w}$, with $C_{\beta w} = C_w + \beta_w$, where β_w is the order toll for order w.

Definition 5 (Order based User Equilibrium)**.** A User Equilibrium is reached when the extra costs made in the network compared to the cheapest path costs of all orders, are payed rationally by all orders. In other words, compared to its cheapest path costs, each order pays the same percentage of extra costs given its cheapest path costs. So $r_{\beta w_1} = r_{\beta w_2}$ $\forall$ $w_1, w_2 \in \mathcal{W}$ (the "fairness" constraint).

In Chapter 4 we will introduce an algorithm to calculate the path tolls, and also show the properties of the found tolls. In Chapter 5 we will do the same for the order tolls. In Chapter 6 we will combine those two toll types and their properties.

4

Tolls on paths

In this chapter, our goal is to find a Path tolled User Equilibrium. Our approach is to first find an SO solution, and then add tolls to paths, which create a new cost scheme for paths. When we offer the STN with the adjusted path costs to the customers, they can selfishy choose routes, and the outcome of their path choice will correspond with our own path to order assignment for finding an SO.

4.1. Constraints for finding path tolls

We assume that customers are not familiar with the path costs of the initial STN in which we want to find an SO solution. We want to find path tolls β_w^p such that when starting with an SO solution, we can adjust the network costs (by adding tolls) in such a way that each customer is satisfied with its route, and thus a UE is achieved. It even should be possible for all customers to let them choose their routes themselves. So we should adjust the path costs in such a way that each path which needs to be traversed in order to obtain an SO solution, will be chosen by the customer.

We only use tolls to obtain both an SO and UE solution, so we do not need to make profit on the tolls. We will discuss the choice of our objective function in Subsection 4.1.1, but we can already state that we want to minimize tolls added to certain path sets.

In order to make sure we do not make profit on the toll system, we require that all tolls payed or received by customers sum up to zero:

$$\sum_{p \in \text{SO solution}} \beta_w^p f_p = 0.$$

So, if there are tolls needed to obtain a UE, there will be one or multiple customers who need to pay toll, as well as there are one or multiple customers who receive toll. This last group thus has a sort of discount on the routes which we want those customers to take. We do not want the toll received by a customer to be higher than the initial path cost (which would mean that a customer does not have to pay, but only receives money for choosing a certain path), so we use the constraints:

$$C_w^p + \beta_w^p \geq 0 \ \forall \ p \in \mathcal{P} \quad \Leftrightarrow \quad \beta_w^p \geq -C_w^p \ \forall \ p \in \mathcal{P}.$$

Our goal is to achieve a UE solution by calculating the path tolls β_w^p, starting from an SO solution. We calculate these tolls with the Nonlinear Programming problem NP-β (as described in Section 4.2 in Algorithm 2 Step 4 at page 42). In Subsection 4.1.1 we will explain our choice for the objective function of the NP-β and we will explain the choice of constraints. In Sections 4.3 up to and including 4.6 we will see how path tolls can be added in the four different cases as described in Section 3.2.

4.1.1. Objective function and choice of constraints

The NP-β consists of an objective function that minimizes the path tolls of a certain path set, and a set of constraints mentioned above. We will elaborate on those constraints in Algorithm 2, but here we will already state the different steps of the algorithm:

Outline Algorithm 2:

1. Create SO problem,
2. Solve SO problem, output: path flows $\underline{f_p}\ \forall\ p \in \mathcal{P}$,
3. Create two lists for each order $w \in \mathcal{W}$: $h_{in,w}$ and $h_{out,w}$,
4. Create NP-β,
5. Solve NP-β, output: path tolls $\beta_w^p\ \forall\ p \in \mathcal{P}$,
6. Create SO-β problem, by adding the path tolls to the path costs of the initial STN,
7. Solve SO-β, output: path flows $\underline{f_p}\ \forall\ p \in \mathcal{P}$.

We have different choices for the objective function of the NP-β, those choices concern minimizing different path sets. Define the set of paths used in the SO solution by

$$h_{in,w} := \left\{p \mid \underline{f_p} > 0,\ p \in \mathcal{P}_w\right\},$$

and the sets of all other paths (which are not in the SO solution) by

$$h_{out,w} := \left\{p \mid \underline{f_p} = 0,\ p \in \mathcal{P}_w\right\}.$$

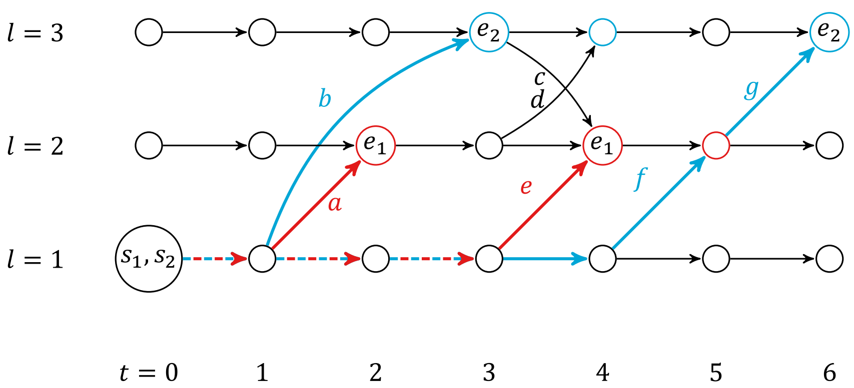


Figure 4.1: STN with two orders, $d_w = 2\ \forall\ w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO. Paths $p \in h_{in,1}$ are denoted in red, paths $p \in h_{in,2}$ in cyan.

To illustrate those path sets, we look at Figure 4.1. There we see two orders both with demand 2 (so this is a Case 2 problem, because $m_{a_i} = 1$ for all arcs a_i between two different locations and $d_w \geq 1\ \forall\ w \in \mathcal{W}$). The SO solution is graphically denoted in red and cyan for order 1 and 2 respectively. We have path sets $\mathcal{P}_1 = \{a, e, bc, f\}$ and $\mathcal{P}_2 = \{b, ad, ag, eg, fg\}$. The SO solution gives us the path sets:

$$h_{in,1} = \{a, e\},\ h_{out,1} = \{bc, f\},\quad h_{in,2} = \{b, fg\},\ \text{ and } h_{out,2} = \{ad, ag, eg, fg\}.$$

We will discuss three possible objective functions for the NP-β in Algorithm 2 Step 4. Notice that we use the absolute values of the path tolls, because we allow both positive and negative tolls.

The three possible objective functions:

- $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} |\beta_w^p|$: minimize the tolls added to paths which are in the SO solution.

- $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p|$: minimize the tolls added to paths which are not in the SO solution.

- $\min \sum_{p \in \mathcal{P}} |\beta_w^p|$: minimize all tolls.

We also discuss whether we want to use the Total-sum-zero constraint $\sum_{p \in \mathcal{P}} \beta_w^p = 0$, which forces the total sum of all tolls added to the STN to be zero. The question arises whether this constraint is necessary to obtain good solutions. By good solutions we mean solutions in which little toll needs to be added to the STN in order to obtain a UE solution.

We first introduce the standard constraints of the NP-β (for Algorithm 2 step 4):

$$\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p \underline{f_p} = 0 \tag{4.1}$$

$$\beta_w^i - \beta_w^j \leq C_w^j - C_w^i \ \forall\ (i,j),\ i \in h_{in,w},\ j \in h_{out,w}\ \forall\ w \in \mathcal{W} \tag{4.2}$$

$$\beta_w^p \geq -C_w^p \ \forall\ p \in \mathcal{P} \tag{4.3}$$

where Constraint 4.1 ensures all tolls on paths used in the SO solution sum up to zero, Constraint 4.2 ensures the paths used in the SO solution for one order, have equal or lower costs than the paths for that order which are not in the SO solution, and Constraint 4.3 ensures no tolled cost can become negative. We will choose our objective function later on. Note that if we work with Case 1, 2 or 3 (as described in Section 3.2), Constraint 4.1 can be replaced (due to the capacity and/or demand constraints) by

$$\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p = 0.$$

This formulation is equivalent because the path flows of paths $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ will always be equal to one due to the capacity and/or demand constraints, and the path flows of paths $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$ will always be equal to zero.
Constraint 4.2 is equivalent to

$$C_w^i + \beta_w^i \leq C_w^j + \beta_w^j \ \forall\ (i,j),\ i \in h_{in,w},\ j \in h_{out,w}\ \forall\ w \in \mathcal{W}.$$

In this equivalent formulation we see the goal of the constraints more clearly.

We use the following two examples to illustrate the influence of the three optional objective functions: $\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} |\beta_w^p|$, $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p|$ and $\sum_{p \in \mathcal{P}} |\beta_w^p|$, and the influence of the optional Total-sum-zero constraint $\sum_{p \in \mathcal{P}} \beta_w^p = 0$.

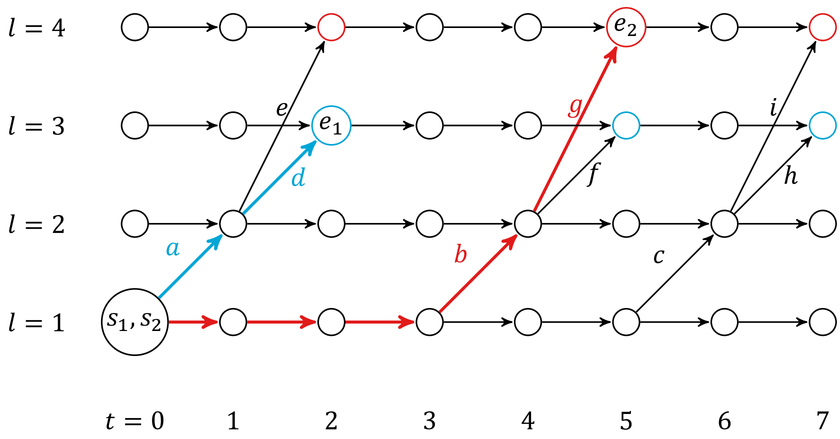


Figure 4.2: STN with two orders,d_w = 1 ∀ $w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO. Paths $p \in h_{in,1}$ are denoted in cyan, paths $p \in h_{in,2}$ in red.

Example 8 (Case 1)**.** The network in Figure 4.2 is of the type Case 1, so $m_{a_i} = 1$ for all arcs a_i between two different locations and $d_w = 1 \ \forall \ w \in \mathcal{W}$. Order 1 travels from location 1 to 3 and order to travels from 1 to 4, $d_1 = d_2 = 1$. We have the following path costs:

Order 1: $C_1^{ad} = 2,\ C_1^{af} = 5,\ C_1^{ah} = 7,\ C_1^{bf} = 5,\ C_1^{bh} = 7,\ C_1^{ch} = 7,$

Order 2: $C_2^{ae} = 2,\ C_2^{ag} = 5,\ C_2^{ai} = 7,\ C_2^{bg} = 5,\ C_2^{bi} = 7,\ C_2^{ci} = 7.$

Solving the corresponding SO problem gives us the solution:

$$\underline{f_{ad}} = 1,\ \underline{f_{bg}} = 1,$$

with path costs of the used paths:

$$C_1^{ad} = 2,\ C_2^{bg} = 5.$$

Then $h_{in,1} = \{ad\}$ and $h_{in,2} = \{bg\}$. We now solve the NP-β with the different objective functions and also observe the effect of the Total-sum-zero constraint: $\sum\limits_{p \in \mathcal{P}} \beta_w^p = 0$:

- Constraint $\sum\limits_{p \in \mathcal{P}} \beta_w^p = 0$:
  - Objective: $\min \sum\limits_{w \in \mathcal{W}} \sum\limits_{p \in h_{in,w}} |\beta_w^p| = \min\left(|\beta_1^{ad}| + \left|\beta_2^{bg}\right|\right) = 0.$

Tolls: $\beta_1^{af} = -3,\quad \beta_1^{ah} = -5,\quad \beta_1^{bf} = -3,\quad \beta_1^{bh} = -5,\quad \beta_1^{ch} = 19,$
$\beta_2^{ae} = 3,\quad \beta_2^{ai} = -2,\quad \beta_2^{bi} = -2,\quad \beta_2^{ci} = -2,$

New path costs: $C_{\beta 1}^{af} = 5 - 3 = 2,\quad C_{\beta 1}^{ah} = 7 - 5 = 2,\quad C_{\beta 1}^{bf} = 5 - 3 = 2,$
$C_{\beta 1}^{bh} = 7 - 5 = 2,\quad C_{\beta 1}^{ch} = 7 + 19 = 26,\quad C_{\beta 2}^{ae} = 2 + 3 = 5,$
$C_{\beta 2}^{ai} = 7 - 2 = 2,\quad C_{\beta 2}^{bi} = 7 - 2 = 5,\quad C_{\beta 2}^{ci} = 7 - 2 = 5.$

The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{ad} = 2,\ C_{\beta 2}^{bg} = 5.$

– Objective: $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = \min\left(\left|\beta_1^{af}\right| + |\beta_1^{ah}| + \left|\beta_1^{bf}\right| + |\beta_1^{bh}| + |\beta_1^{ch}| + |\beta_2^{ae}| + |\beta_2^{ag}| + |\beta_2^{ai}| + |\beta_2^{bi}| + |\beta_2^{ci}|\right) = 0.$

Tolls:	$\beta_1^{ad} = 3,$	$\beta_2^{bg} = -3,$
New path costs:	$C_{\beta 1}^{ad} = 2 + 3 = 5,$	$C_{\beta 2}^{bg} = 5 - 3 = 2.$

The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{ad} = 5,\ C_{\beta 2}^{bg} = 2.$

– Objective: $\min \sum_{p \in \mathcal{P}} |\beta_w^p| = \min\left(|\beta_1^{ad}| + \left|\beta_1^{af}\right| + |\beta_1^{ah}| + \left|\beta_1^{bf}\right| + |\beta_1^{bh}| + |\beta_1^{ch}| + |\beta_2^{ae}| + |\beta_2^{ag}| + |\beta_2^{ai}| + \left|\beta_2^{bg}\right| + |\beta_2^{bi}| + |\beta_2^{ci}|\right) = 6.$

Tolls:	$\beta_1^{ch} = -3,$	$\beta_2^{ae} = 3,$
New path costs:	$C_{\beta 1}^{ch} = 7 - 3 = 4,$	$C_{\beta 2}^{ae} = 2 + 3 = 5.$

The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{ad} = 2,\ C_{\beta 2}^{bg} = 5.$

- No constraint $\sum_{p \in \mathcal{P}} \beta_w^p = 0$:

– Objective: $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} |\beta_w^p| = \min\left(|\beta_1^{ad}| + \left|\beta_2^{bg}\right|\right) = 0$, total sum of tolls is $\sum_{p \in \mathcal{P}} \beta_w^p = -24.$

Tolls: $\beta_1^{af} = -3,\quad \beta_1^{ah} = -5,\quad \beta_1^{bf} = -3,\quad \beta_1^{bh} = -5,\quad \beta_1^{ch} = -5,$
$\beta_2^{ae} = 3,\quad \beta_2^{ai} = -2,\quad \beta_2^{bi} = -2,\quad \beta_2^{ci} = -2,$

New path costs: $C_{\beta 1}^{af} = 5 - 3 = 2,\quad C_{\beta 1}^{ah} = 7 - 5 = 2,\quad C_{\beta 1}^{bf} = 5 - 3 = 2,$
$C_{\beta 1}^{bh} = 7 - 5 = 2,\quad C_{\beta 1}^{ch} = 7 - 5 = 2,\quad C_{\beta 2}^{ae} = 2 + 3 = 5,$
$C_{\beta 2}^{ai} = 7 - 2 = 5,\quad C_{\beta 2}^{bi} = 7 - 2 = 5,\quad C_{\beta 2}^{ci} = 7 - 2 = 5.$

The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{ad} = 2,\ C_{\beta 2}^{bg} = 5.$

– Objective: $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$, total sum of tolls is $\sum_{p \in \mathcal{P}} \beta_w^p = 0.$

Tolls:	$\beta_1^{ad} = 3,$	$\beta_2^{bg} = -3,$
New path costs:	$C_{\beta 1}^{ad} = 2 + 3 = 5,$	$C_{\beta 2}^{bg} = 5 - 3 = 2.$

The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{ad} = 5,\ C_{\beta 2}^{bg} = 2.$

– Objective: $\min \sum_{p \in \mathcal{P}} |\beta_w^p| = 3$, total sum of tolls is $\sum_{p \in \mathcal{P}} \beta_w^p = 3.$

Toll:	$\beta_2^{ae} = 3,$
New path cost:	$C_{\beta 2}^{ae} = 2 + 3 = 5.$

The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{ad} = 2,\ C_{\beta 2}^{bg} = 5.$

△

We see that using the objective $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} |\beta_w^p|$ is highly inefficient. Inefficiency here means that given the SO solution, if there are a lot of cheaper alternative paths for orders $w \in \mathcal{W}$ (so $C_w^q \leq C_w^p$ for $p \in h_{in,w}$, $q \in h_{out,w}$), we need to add tolls to all those paths in order to

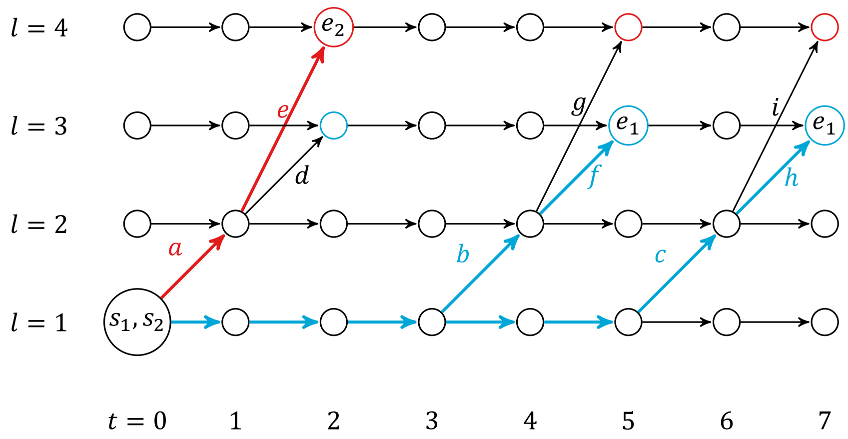


Figure 4.3: STN with two orders, $d_1 = 2$, $d_2 = 1$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO. Paths $p \in h_{in,1}$ are denoted in cyan, paths $p \in h_{in,2}$ in red.

obtain a UE. We need to do this regardless of the constraint $\sum_{p \in \mathcal{P}} \beta_w^p = 0$. In the next example we will thus only consider the other two objective functions.

Because in order to maintain the paths costs for the used paths, one needs to adjust all costs of the alternative paths for all orders. When increasing the demand for order 1 from the previous example to 2, we can see what more pressure on the network does for the solutions.

Example 9 (Case 2)**.** This example is of the type Case 2, because $m_{a_i} = 1$ for all arcs a_i between two different locations and $d_w \geq 1 \ \forall \ w \in \mathcal{W}$. We have the same STN as in the previous example, now with $d_1 = 2$, $d_2 = 1$. Solving the SO problem gives the following solution, which is also displayed in Figure 4.3:

$$\underline{f_{bf}} = 1,\ \underline{f_{ch}} = 1,\ \underline{f_{ae}} = 1,$$
$$C_1^{bf} = 5,\ C_1^{ch} = 7,\ C_2^{ae} = 2.$$

Then $h_{in,1} = \{bf,\ ch\}$, $h_{in,2} = \{ae\}$. We now solve the NP-β with the different objective functions and also observe the effect of the Total-sum-zero constraint, by removing this constraint from the NP-β and compare the values of $\sum_{p \in \mathcal{P}} \beta_w^p$.

- Constraint $\sum_{p \in \mathcal{P}} \beta_w^p = 0$:
  - Objective: $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 6$.

    Tolls: $\beta_1^{ad} = 3$, $\beta_1^{bh} = -2$, $\beta_1^{ch} = -3$, $\beta_2^{ae} = 3$, $\beta_2^{ai} = -1$,

    New path costs: $C_{\beta 1}^{ad} = 2 + 3 = 5$, $C_{\beta 1}^{bh} = 7 - 2 = 5$, $C_{\beta 1}^{ch} = 7 - 3 = 4$,
    $C_{\beta 2}^{ae} = 2 + 3 = 5$, $C_{\beta 2}^{ai} = 7 - 1 = 6$.

    The costs of the paths used in the SO solution are thus: $C_{\beta 1}^{bf} = 5$, $C_{\beta 1}^{ch} = 4$, $C_{\beta 2}^{ae} = 5$.
  - Objective: $\min \sum_{p \in \mathcal{P}} |\beta_w^p| = 10$.

    Tolls: $\beta_1^{ad} = 3$, $\beta_1^{bh} = -2$, $\beta_1^{ch} = -2$, $\beta_2^{ae} = 2$, $\beta_2^{ci} = -1$,

    New path costs: $C_{\beta 1}^{ad} = 2 + 3 = 5$, $C_{\beta 1}^{bh} = 7 - 2 = 5$, $C_{\beta 1}^{ch} = 7 - 2 = 5$,
    $C_{\beta 2}^{ae} = 2 + 2 = 4$, $C_{\beta 2}^{ci} = 7 - 1 = 6$.

The costs of the paths used in the SO solution are thus: $C^{bf}_{\beta 1} = 5,\ C^{ch}_{\beta 1} = 5,\ C^{ae}_{\beta 2} = 4.$

- No constraint $\sum_{p\in\mathcal{P}} \beta^p_w = 0$:
  - Objective: $\min \sum_{w\in\mathcal{W}} \sum_{p\in h_{out,w}} |\beta^p_w| = 3$, total sum of tolls is $\sum_{p\in\mathcal{P}} \beta^p_w = 3$.

$$\begin{array}{lll} \text{Tolls:} & \beta_1^{ad} = 3, \quad \beta_1^{ch} = -2, & \beta_2^{ae} = 3, \quad \beta_1^{bf} = -1, \\ \text{New path costs:} & C^{ad}_{\beta 1} = 2+3 = 5, & C^{ch}_{\beta 1} = 7-2 = 5, \\ & C^{ae}_{\beta 2} = 2+3 = 5, & C^{bf}_{\beta 2} = 5-1 = 4. \end{array}$$

  So $C^{bf}_{\beta 1} = 4,\ C^{ch}_{\beta 1} = 5,\ C^{ae}_{\beta 2} = 5.$

  - Objective: $\min \sum_{p\in\mathcal{P}} |\beta^p_w| = 7$, total sum of tolls is $\sum_{p\in\mathcal{P}} \beta^p_w = 7$.

$$\begin{array}{lll} \text{Toll:} & \beta_1^{ad} = 5, & \beta_1^{af} = 2, \\ \text{New cost:} & C^{ad}_{\beta 1} = 2+5 = 8, & C^{af}_{\beta 1} = 5+2 = 7. \end{array}$$

  The costs of the paths used in the SO solution are thus: $C^{bf}_{\beta 1} = 5,\ C^{ch}_{\beta 1} = 7,\ C^{ae}_{\beta 2} = 2.$

$\triangle$

We can see in the previous examples, that adding the Total-sum-zero constraint does not give better solutions when the goal is to add as little as toll necessary, which we want to achieve. So we will no longer use this Total-sum-zero constraint.
From now on we will minimize the tolls added to paths which are not in the SO solution, so take we take as the objective function:

$$\sum_{w\in\mathcal{W}} \sum_{p\in h_{out,w}} |\beta^p_w|.$$

4.2. Generic algorithm for finding path tolls

We start determining tolls by solving the SO, and then add tolls to the path costs in the STN such that no customer is harmed by that solution and thus a UE is reached. The sum of all tolls payed and received by all customers has to be zero. The way of finding tolls which give us a UE solution in a initial SO problem is described in Algorithm 2, which is partly based on the solution algorithms used by Hearn and Ramana [13] and Jiang and Mahmassani [14]. Hearn and Ramana work with a toll pricing framework:

1. Solve the SO problem to obtain an optimal solution $\boldsymbol{x}^*$.
2. Define the toll set.
3. Define and optimize an objective function over the toll set.

Note that in our case we can use the Robinhood objective as in Equation 2.32.
Jiang and Mahmassani have a Unifying Solution Algorithm for both discrete and continuous Value of Time:

0. Initialization.
1. Traffic Simulation.
2. Generating Path Set: Find cheapest path for each user.
3. Update Path Assignment: For each user, update path flow by using Method of Successive Averages.
4. Convergence Check.

The difference with the framework of Hearn and Ramana is that we do not define the toll set, because in our approach there is no need to obtain this total set. The difference with Jiang and Mahmassani is that we apply tolls on paths instead of updating path assignment.

Recall, to calculate the path tolls, we divide all paths into two sets, one with all paths which are used in the SO solution: $h_{in,w}$, and one with all other paths (which are not in the SO solution): $h_{out,w}$. We define these sets as follows:

$$h_{in,w} = \left\{p \mid \underline{f_p} > 0,\ p \in \mathcal{P}_w\right\}, \quad h_{out,w} = \left\{p \mid \underline{f_p} = 0,\ p \in \mathcal{P}_w\right\}$$

Algorithm 2 Calculating path tolls

1: Create SO problem:

$$\begin{aligned} \min \ & \sum_{p \in \mathcal{P}} C_w^p f_p \\ \text{s.t.} \ & x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \ \forall\ a \in \mathcal{A} \\ & \sum_{p \in \mathcal{P}_w} f_p = d_w \ \forall\ w \in \mathcal{W} \\ & x_a \leq m_a \ \forall\ a \in \mathcal{A} \\ & f_p \in \mathbb{N}_0 \ \forall\ p \in \mathcal{P} \\ & x_a \in \mathbb{N}_0 \ \forall\ a \in \mathcal{A} \end{aligned} \tag{4.4}$$

2: Solve SO problem, output: path flow vector $\underline{\boldsymbol{f}}$.

3: Create two lists for each order w: $h_{in,w} = \left\{p \mid \underline{f_p} > 0,\ p \in \mathcal{P}_w\right\}$, $h_{out,w} = \left\{p \mid \underline{f_p} = 0,\ p \in \mathcal{P}_w\right\}$.

4: Create NP-β:

$$\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} \left|\beta_w^p\right| \tag{4.5}$$

$$\text{s.t.} \sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p \underline{f_p} = 0 \tag{4.6}$$

$$\beta_w^i - \beta_w^j \leq C_w^j - C_w^i \ \forall\ (i,j),\ i \in h_{in,w},\ j \in h_{out,w} \ \forall\ w \in \mathcal{W} \tag{4.7}$$

$$\beta_w^p \geq -C_w^p \ \forall\ p \in \mathcal{P} \tag{4.8}$$

where Constraint 4.6 ensures the total toll sum of the chosen paths to be zero, Constraint 4.7 ensures that the paths in the SO solutions are the ones with cheapest costs $C_{\beta w}^p$ and Constraint 4.8 ensures no path can have a negative $C_{\beta w}^p$ cost.

The NP-β in Algorithm 2 Step 4 is non-linear, which makes this problem hard to solve. We therefore use the equivalent linear formulation of the problem:

$$\begin{aligned}
\min\ & \sum_{w\in\mathcal{W}} \sum_{p\in h_{out,w}} \gamma_w^p \\
\text{s.t.}\ & \sum_{w\in\mathcal{W}} \sum_{p\in h_{in,w}} \beta_w^p \underline{f_p} = 0 \\
& \beta_w^i - \beta_w^j \leq C_w^j - C_w^i \ \forall\ (i,j),\ i \in h_{in,w},\ j \in h_{out,w}\ \forall\ w \in \mathcal{W} \\
& \beta_w^p \geq -C_w^p\ \forall\ p \in \mathcal{P} \\
& \beta_w^p \leq \gamma_w^p\ \forall\ p \in \mathcal{P} && (4.9)\\
& -\beta_w^p \leq \gamma_w^p\ \forall\ p \in \mathcal{P} && (4.10)\\
& \gamma_w^p \geq 0\ \forall\ p \in \mathcal{P} && (4.11)
\end{aligned}$$

where γ_w^p replaces the absolute value variable $|\beta_w^p|$ with Constraints 4.9-4.11.

5: Solve NP-β, output: β_w^p.

6: Add tolls β_w^p to the SO problem, SO-β:

$$\begin{aligned}
\min\ & \sum_{p\in\mathcal{P}} \left(C_w^p + \beta_w^p\right) f_p \\
\text{s.t.}\ & x_a = \sum_{p\in\mathcal{P}} \delta_{ap} f_p\ \forall\ a \in \mathcal{A} \\
& \sum_{p\in\mathcal{P}_w} f_p = d_w\ \forall\ w \in \mathcal{W} \\
& x_a \leq m_a\ \forall\ a \in \mathcal{A} \\
& f_p \in \mathbb{N}_0\ \forall\ p \in \mathcal{P} \\
& x_a \in \mathbb{N}_0\ \forall\ a \in \mathcal{A}
\end{aligned}$$

7: Solve SO-β problem, output path flow vector $\underline{\boldsymbol{f}}$.

The desired outcome of Algorithm 2, is that the solution to the SO-β problem is equal to the initial SO problem. The resulting path costs are the only costs that are showed to the customers, so the customers do not have any knowledge about the initial STN and those path costs. We will consider the problem of solving an STN in different cases, as stated in Section 3.2.

4.3. Case 1

Case 1 is the situation in which $d_w = 1\ \forall\ w \in \mathcal{W}$, and $m_a = 1\ \forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
We illustrate the correctness of Algorithm 2, by solving a number of examples in Subsection 4.3.1. In Subsection 4.3.2 we will derive properties of the path tolls.

4.3.1. Case 1: Examples

In Example 10 and Example 11 we will show the outcome of the algorithm on the problems and have a small discussion on the tolls found the algorithm.

Example 10. We use the STN as in Figure 3.1, again showed in Figure 4.4. Recall that all arcs have cost 1 and all diagonal arcs have capacity constraints of 1. We have two orders: $\mathcal{W} = \{1, 2\}$.
The SO solution is given in Figure 4.4, with s_w denoting the start point and e_w denoting the end point for order w.

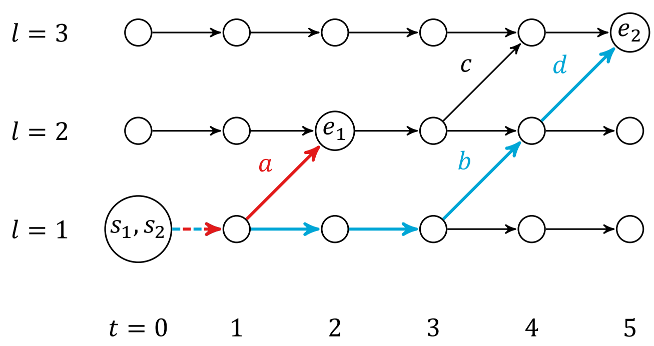


Figure 4.4: STN with two orders with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

We have path costs

$$C_1^a = 2,\ C_1^b = 4,\ C_2^{ac} = 4,\ C_2^{ad} = 5 \text{ and } C_2^{bd} = 5,$$

and the path sets following from the SO solution as obtained in Algorithm 2 in Step 3:

$$h_{in,1} = \{a\},\ h_{in,2} = \{bd\},\ h_{out,1} = \{b\},\ h_{out,2} = \{ac, ad\}.$$

We see that due to capacity constraints, order 2 is not able to choose his best route in the SO and has to take path bd with cost 5.
The tolls given by Algorithm 2 are

$$\beta_1^a = 1,\ \beta_2^{bd} = -1,$$

so all tolls on paths $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$ are zero and so is the objective value of the NP-β.
We will take a closer look at how we can look at these tolls and adjusted costs: The paths a and bd are in the SO solution with a total objective value of 7.
We now can set bounds on the toll: one could say that the toll of order 1 can be at most 2, so $\beta_1^a < 2$, which gives a lower bound of -2 to the toll of order 2: $\beta_2^{bd} > -2$. We set those bounds because if the toll of order 1 on path a gets higher than 2, it is more beneficial for order 1 to take its alternative route b. For order 2, the smallest toll can be -1, so $\beta_2^{bd} > -1$, which gives an upper bound of 1 to the toll of order 1: $\beta_1^a < 1$. If we do not set those bounds, order 2 adjusted path cost will be cheaper than traveling via its cheapest path in the initial network.
But if we look at the tolled ratios $r_{\beta w} = \dfrac{C_w + \beta_w}{k_w}$, where we compare the adjusted path cost to the cheapest path cost, we see that a toll of 1 for order 1 is more disadvantageous than the initial SO solution was for order 2 (see Table 4.1).

When we assume that all customers do know the path costs of the initial network (contrary to what we assume in this Section), order 2 will not accept path bd as its route as long as $\beta_2^{bd} > -1$, because then $C_{\beta 2}^{bd} = 5 + \beta_2^{bd} >= C_2^{ac} = k_2$ and thus path ac is still a cheaper route to take. A negative toll of value $\beta_2^{bd} = -1$ will be satisfying for order 2, because then $C_{\beta 2}^{bd} = C_2^{bd} + \beta_2^{bd} = 5 - 1 = 4 = C_2^{ac} = k_2$, and thus there is no difference in both optional paths costs.
Order 1 can accept paying the toll of $\beta_1^a = 1$, because then the cost of path a will still be less than its

Table 4.1: Solutions for Figure 4.4: Ratios $r_{\beta w} = (C_w + \beta_w)/k_w$ per toll

	Order 1	$C^a_{\beta 1}$	r_1	Order 2	$C^{bd}_{\beta 2}$	r_2
k_w	2			4		
C^p_w	2		1	5		1.25
β^p_w	0.5	2.5	1.25	-0.5	4.5	1.125
	1	3	1.5	-1	4	1
	2	4	2	-2	3	0.75

alternative path b which has cost $C^b_1 = 4$ (and $C^a_{\beta 1} = C^a_1 + \beta^a_1 = 2 + 1 = 3 < 4$). The question is however, to what extent order 1 is willing to pay 'voluntarily' to order 2, to obtain a User Equilibrium.

Now assume the customers do not have any knowledge of the path costs of the initial STN. Then we can offer a choice to order 1 of choosing the path a with toled cost $C^a_{\beta 1} = 2 + 1 = 3$ or path b with tolled cost $C^b_{\beta 1} = 5 - 1 = 4$, and offer to order 2 three paths, with ac and bd having both tolled cost 4 and ad having cost 5. Then they can both choose their most favorable route (a and bd respectively) without harming the other order.
But what will happen if we use tolls $|\beta^p_w| < 1 \ \forall\ p \in \mathcal{P}$, and we assume we only add tolls to paths chosen in the SO solution? Take for example path tolls $\beta_1 = \beta^a_1 = 0.5$ and $\beta_2 = \beta^{bd}_1 = -0.5$. Then order 1 gets to choose between a path costs $C^a_{\beta 1} = 2.5$ and $C^b_{\beta 1} = 4$, while order 2 has to choose between $C^{ac}_{\beta 2} = 4$, $C^{ad}_{\beta 2} = 5$ and $C^{bd}_{\beta 2} = 4.5$. Then both orders will choose their cheapest path, but those paths both contain the first diagonal arc going from location 1 to 2. So this solution will not give the desired choices of the orders.

The best solution of the NP-β is indeed the solution as obtained from Algorithm 2 Step 5:

$$\beta^a_1 = 1,\ \beta^b_1 = 0,\ \beta^{ac}_2 = 0,\ \beta^{ad}_2 = 0,\ \beta^{bd}_2 = -1$$

and with those tolls we obtain the path costs:

$$C^a_{\beta 1} = 3,\ C^b_{\beta 1} = 4,\ C^{ac}_{\beta 2} = 4,\ C^{ad}_{\beta 2} = 5 \text{ and } C^{bd}_{\beta 2} = 4,$$

so both orders can travel via their cheapest paths, so both an SO and a UE are obtained. △

Is it sufficient to only add tolls to paths that are in the SO solution, or do we need to also take the other paths into account? We will show in the following example that adding tolls to paths $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$ can also be useful.

Example 11. We changed the network from Example 10 by changing two diagonal arcs starting at $t = 3$ in Figure 4.4 to $t = 2$. We will now observe which toll we need to add to obtain a UE which equals the initial SO. Again we have two orders: $\mathcal{W} = \{1, 2\}$. The SO solution is given in Figure 4.5, with s_w and e_w denoting the start end point of order w, respectively.
We have path costs

$$C^a_1 = 2,\ C^b_1 = 3,\ C^{ac}_2 = 3,\ C^{ad}_2 = 5 \text{ and } C^{bd}_2 = 5,$$

and the path sets following from the SO solution:

$$h_{in,1} = \{b\},\ h_{in,2} = \{ac\},\ h_{out,1} = \{a\},\ h_{out,2} = \{ad, bd\}.$$

The solution given by Algorithm 2 Step 5 is

$$\beta^b_1 = -1,\ \beta^{ac}_2 = 1.$$

We see that with this tolls, order 1 can choose between path a with cost $C^a_{\beta 1} = 1$ and path b with cost $C^b_{\beta 1} = 1$, so it makes no difference to order 1 which path is chosen. Order 2 can choose between

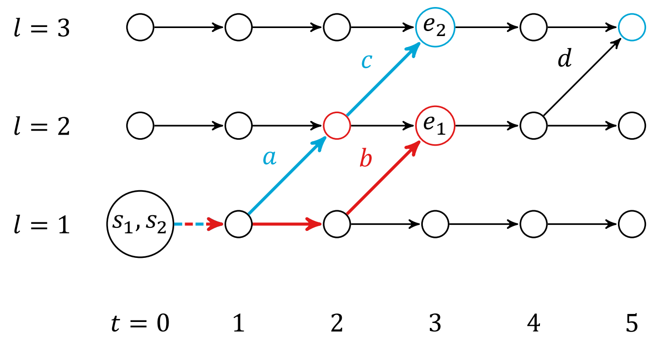


Figure 4.5: STN with two orders with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

paths ac, ad and bd, where the path cost of path ac is still cheaper than the other two path costs: $C^{ac}_{\beta 2} = 4 < 5 = C^{ad}_{\beta 2} = C^{bd}_{\beta 2}$.
When we want to add tolls to used paths only, we obtain the following bounds on the path tolls: order 1 does not travel via its cheapest route a, so he wants his route b cheaper with cost 1, $\beta^b_1 < -1$, while order 2 can pay 2 before his alternative path bd becomes cheaper than the initial cheapest path ac: so $\beta^{ac}_2 < 2$. We see that the given toll solution satisfy those bounds.

Another valid solution of the NP-β could be adding only toll to path a: $\beta^a_1 = 1$, $\beta^b_1 = 0$, $\beta^{ac}_2 = 0$, $\beta^{ad}_2 = 0$, and $\beta^{bd}_2 = 0$.
With those tolls we obtain the path costs: $C^a_{\beta 1} = 3$, $C^b_{\beta 1} = 3$, $C^{ac}_{\beta 2} = 3$, $C^{ad}_{\beta 2} = 5$ and $C^{bd}_{\beta 2} = 5$ and we see that the paths chosen in the SO solution now indeed are the cheapest paths for the customers.
But note that this solution does not minimize the objective function of the NP-β, because

$$\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta^p_w| = |\beta^a_1| + |\beta^{ad}_2| + |\beta^{bd}_2| = 1,$$

which is higher than the objective value of the solution given by Algorithm 2 Step 5 (which is 0). △

We can conclude from the two previous examples that for instances with two orders, the gap between the cheapest path and the initial SO solution plus tolls has to be zero to force the customers to take the paths that satisfy the SO solution.

4.3.2. Case 1: Finding properties of path tolls

In this section we will show some properties of path tolls in Case 1. In this Case 1, we have for all p, a: $f_p \in \{0, 1\}$, $x_a \in \{0, 1\}$. After solving the SO problem we can define

$$h_{in,w} = \left\{p \mid \underline{f_p} = 1,\ p \in \mathcal{P}_w\right\}, \quad |h_{in,w}| = 1\ \forall\ w \in \mathcal{W},$$
$$h_{out,w} = \left\{p \mid \underline{f_p} = 0,\ p \in \mathcal{P}_w\right\}.$$

In this case the flow of an order is 1, so we can write

$$\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta^p_w \underline{f_p} = \sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta^p_w = 0.$$

We use the NP-β as in Algorithm 2 Step 4.

We start with an observation of the output types of path tolls.

Observation 1. *If a solution to the NP-β for path tolls exist, it is of one of the following forms:*

1. $\beta_w^p = 0\ \forall\ p \in \mathcal{P}$,
2. $\beta_w^p \neq 0$ *for at least two paths* $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ *and* $\beta_w^p = 0\ \forall\ p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$,
3. $\beta_w^p \neq 0$ *for at least one path* $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$ *and* $\beta_w^p = 0\ \forall\ p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$,
4. $\beta_w^p \neq 0$ *for at least two paths* $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ *and* $\beta_w^p \neq 0$ *for at least one path* $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$.

When $\beta_w^p = 0\ \forall\ p \in \mathcal{P}$, it means that the SO solution is equal to the UE solution in the initial problem. The question is, if it is sufficient to have solutions of the form 2 and 3, or if there problems in which it is necessary to have solutions of the 4th form.

To illustrate the solutions above, we show the following example.

Example 12. We use the STN of Figure 4.5 on page 46. The SO solution is given for the two demand pairs, $\underline{f_b} = 1$ for order 1 and $\underline{f_{ac}} = 1$ for order 2. We create the NP-β as in Algorithm 2 Step 4 and solve it.
We can obtain several outcomes for this STN:

2. $\beta_2^{ac} = 1,\ \beta_1^b = -1\ \Rightarrow\ C_{\beta 1}^a = 2,\ C_{\beta 1}^b = 3 - 1 = 2,\ C_{\beta 2}^{ac} = 3 + 1 = 4,\ C_{\beta 2}^{bd} = 5,$
3. $\beta_1^a = 1\ \Rightarrow\ C_{\beta 1}^a = 2 + 1 = 3,\ C_{\beta 1}^b = 3,\ C_{\beta 2}^{ac} = 3,\ C_{\beta 2}^{bd} = 5,$
4. A combination of the solutions 2 and 3 is also a valid solution:
$\beta_2^{ac} = 1,\ \beta_1^b = -1,\ \beta_1^a = 1,\ \Rightarrow C_{\beta 1}^a = 2 + 1 = 3,\ C_{\beta 1}^b = 3 - 1 = 2,\ C_{\beta 2}^{ac} = 3 + 1 = 4,\ C_{\beta 2}^{bd} = 5.$

We see that a solution of the first form ($\beta_w^p = 0\ \forall\ p \in \mathcal{P}$) is not a solution in this example. Based on the objective function $\min \sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p|$ we use solving the NP-β, solution 2 is the best solution. △

We will now investigate the path toll solutions by increasing order size.

Case 1 with 1 order
If we have an STN with only 1 order, then the SO solution will always be equal to the UE solution, more specifically, the SO solution will always be traveling by the cheapest path of the order. An example is given in Figure 4.6.

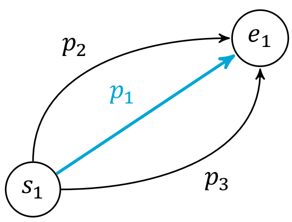


Figure 4.6: One order with multiple paths

Case 1 with two orders
When multiple orders have to be transported in the same time frame, paths for the different orders can have joint arcs. When this is the case, it can influence the SO, due to the capacity constraint that only one container can travel an arc ($m_a = 1$). We can see this in Figure 4.7.

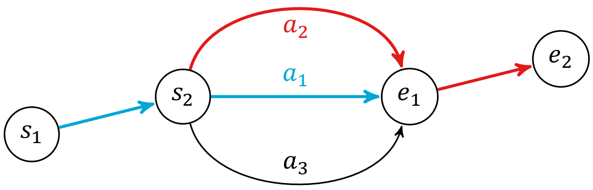


Figure 4.7: Network with two orders with multiple paths

Here we have two orders, with three arcs between the start point of order 2, which is s_2, and the end point of order 1, which is e_1: the arcs we denote by a_1, a_2 and a_3. Assume the following holds for the arc costs: $c_{a_1} \leq c_{a_2} \leq c_{a_3}$. Then for both orders arc a_2 is in the cheapest path, but because $m_a = 1$ for all arcs, only one order can traveling via a_2. We see that in this example it is possible in the SO solution for one order to travel via its cheapest path.

We show that it is sometimes better to let no order travel via its cheapest path in order to obtain an SO solution. We show the following example.

Example 13. We use the graph in Figure 4.8.

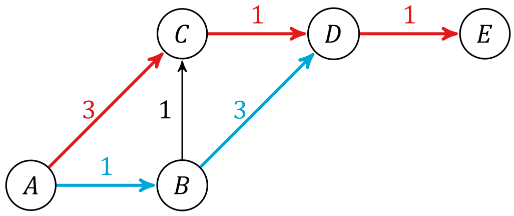


Figure 4.8: Network with order OD-pairs AD and AE.

Suppose all arcs in this network have capacity $m_a = 1$ and there are two orders, both transporting one container, one traveling from A to D and one from A to E. The costs of the arcs are given in Figure 4.8. For order 1 there are three paths: ABD, $ABCD$, ACD, and for order 2 there are also three paths: $ABDE$, $ABCDE$, $ACDE$. The path costs are $C_1^{ABD} = C_1^{ACD} = 4$, $C_1^{ABCD} = 3$, $C_2^{ABDE} = C_2^{ACDE} = 5$, $C_2^{ABCDE} = 4$. An SO solution in this example is order 1 traveling via ABD and order 2 via $ACDE$, although then both orders cannot travel via their cheapest path.

We can also translate this graph into an STN: see Figure 4.9.

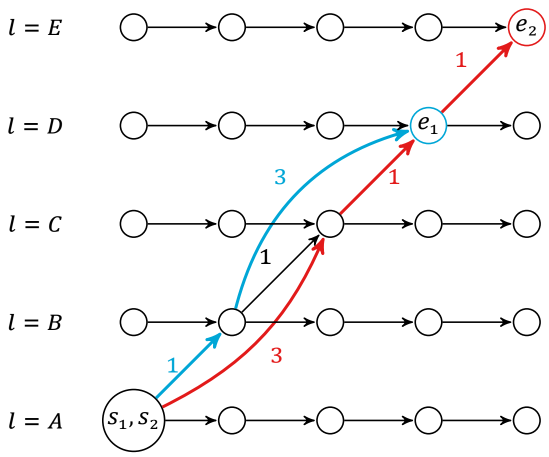


Figure 4.9: STN with costs c_a on diagonal arcs, $m_a = 1$ for all arcs between two different locations, $m_a = \infty$ otherwise. The denoted solution is SO.

In this network it is sufficient to add toll to the unused arc BC to make sure no order will travel via this path. Adding an arc toll $\beta^{BC} = 1$ is sufficient, because then for each order all paths have the same costs. This arc based toll is equivalent to adding toll to the paths $ABCD$ and $ABCDE$: $\beta_1^{ABCD} = 1$, $\beta_2^{ABCDE} = 1$, respectively. When we solve the NP-β for this problem, we want to minimize the path tolls of unused paths. But in this case no order can travel via its cheapest path, so we are forced to increase the path costs of those cheapest paths $ABCD$ and $ABCDE$, which both contain arc BC. △

Assume the SO problem in the STN with capacity $m_a = 1$ for all traveling arcs $a = (l_1, l_2) \in \mathcal{A}$ with $l_1 \neq l_2$, has an optimal solution.
Define

$$h_{in,w} := \left\{p \mid \underline{f_p} > 0,\ p \in \mathcal{P}_w\right\}, \quad h_{out,w} := \left\{p \mid \underline{f_p} = 0,\ p \in \mathcal{P}_w\right\}\ \forall\ w \in \mathcal{W}, \quad \left|h_{in,w}\right| = 1\ \forall\ w \in \mathcal{W}.$$

We will show in general that an SO solution does not always contain cheapest paths for some orders, as we showed for the STN in Figure 4.9.
We define $s^j \in \mathcal{S}$ a feasible solution, with $s^j = (\boldsymbol{x}^j, \boldsymbol{f}^j)$ and $\mathcal{S}$ the set of all feasible solutions. We denote the SO solution by $s^1 = \left(\underline{\boldsymbol{x}^1}, \underline{\boldsymbol{f}^1}\right) \in \mathcal{S}$, and $s^j = \left(\boldsymbol{x}^j, \boldsymbol{f}^j\right) \in \mathcal{S}$, $j \in \{2, 3, \dots, |\mathcal{S}| - 1, |\mathcal{S}|\}$ are all other feasible solutions.

For the following propositions, we will define the following: $\mathcal{W} = \{1, 2\}$, $d_1 = d_2 = 1$, and the path set for both orders:

$$\mathcal{P}_1 = \{r_1, r_2, \dots, r_n\},\ \mathcal{P}_2 = \{q_1, q_2, \dots, q_m\}.$$

So $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 = \{r_1, \dots, r_n, q_1, \dots, q_m\}$. Assume the paths are ordered such that

$$C_1^{r_1} \leq C_1^{r_2} \leq \dots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \dots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

and assume the SO solution value is $C_1^{r_a} + C_2^{q_b}$, so the path vector $\boldsymbol{f}^1$ of s^1 contains elements $f_{r_a}^1 = 1,\ f_{q_b}^1 = 1,\ f_p^1 = 0\ \forall\ p \in \mathcal{P} \setminus \{r_a, q_b\}$. Then

$$C_1^{r_a} + C_2^{q_b} \leq C_1^{r_k} + C_2^{q_l}\ \forall\ s^j = \left(\boldsymbol{x}^j, \boldsymbol{f}^j\right) \in S \setminus \{s^1\} \text{ with } f_{r_k}^j = 1,\ f_{q_l}^j = 1,\ f_p^j = 0\ \forall\ p \in \mathcal{P} \setminus \{r_k, q_l\}.$$

Which solutions on the tolls do we obtain?

- If for order 1: $r_1 \in h_{in,1}$ and for order 2: $q_1 \in h_{in,2}$ (so both orders are traveling via their cheapest path), then no tolls need to be added: $\beta_w^p = 0\ \forall\ p \in \mathcal{P}$, see Proposition 2 and Figure 4.10.

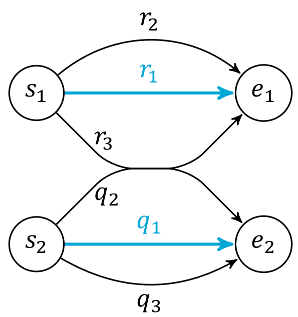
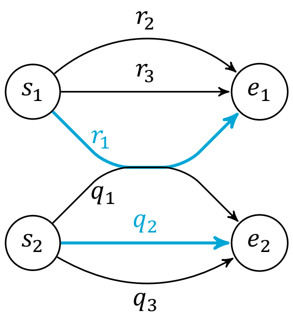
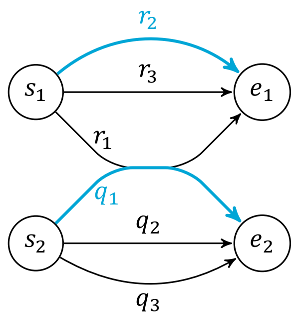


Figure 4.10: Two orders with multiple paths: situation as in Proposition 2, SO solution in cyan.

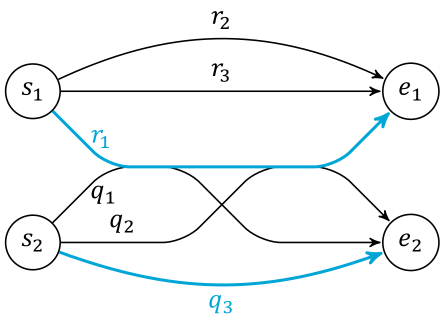
- If for order 1: $r_1 \in h_{in,1}$ and for order 2: $q_l \in h_{in,2}$, $2 \leq l \leq m$ with path costs such that $C_1^{r_1} + C_2^{q_l} \leq C_1^{r_2} + C_2^{q_1}$ or
if for order 1: $r_k \in h_{in,1}$, $2 \leq k \leq n$ and for order 2: $q_1 \in h_{in,2}$ with $C_1^{r_k} + C_2^{q_1} \leq C_1^{r_1} + C_2^{q_2}$, then it is sufficient to add tolls on the used paths only, see Proposition 5 and Figure 4.11a and 4.11b (in Figure 4.11a we have paths costs such that $C_1^{r_1} + C_2^{q_l} \leq C_1^{r_2} + C_2^{q_1}$ with $l = 2$ always holds, and in Figure 4.11b we have path costs such that $C_1^{r_k} + C_2^{q_1} \leq C_1^{r_1} + C_2^{q_2}$ with $k = 2$ always holds). In Figure 4.11c and Figure 4.11d we see more examples of Proposition 5: in Figure 4.11c we assume $C_1^{r_1} + C_2^{q_l} \leq C_1^{r_2} + C_2^{q_1}$ and in Figure 4.11d we assume $C_1^{r_2} + C_2^{q_1} \leq C_1^{r_1} + C_2^{q_3}$.



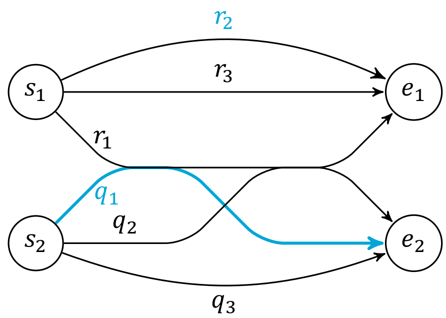
(a) SO solution in cyan with $C_1^{r_1} + C_2^{q_2} \leq C_1^{r_2} + C_2^{q_1}$.



(b) SO solution in cyan with $C_1^{r_1} + C_2^{q_2} \geq C_1^{r_2} + C_2^{q_1}$.



(c) SO Solution in cyan with $C_1^{r_1} + C_2^{q_3} \leq C_1^{r_2} + C_2^{q_1}$.



(d) SO Solution in cyan with $C_1^{r_1} + C_2^{q_3} \geq C_1^{r_2} + C_2^{q_1}$.

Figure 4.11: Two orders with multiple paths: situation as in Proposition 5.

- If for order 1: $r_k \in h_{in,1}$, $2 \leq k \leq n$ and for order 2: $q_l \in h_{in,2}$, $2 \leq l \leq m$, then tolls on the unused paths with lower path cost are necessary (so tolls on paths $p \in h_{out,1}$ with $C_1^p \leq C_1^{r_k}$ and paths $p \in h_{out,2}$ with $C_2^p \leq C_2^{q_l}$), see Proposition 3 and Figure 4.12.

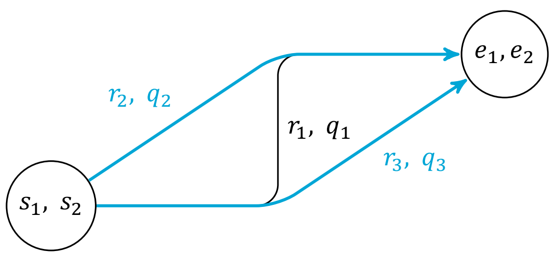


Figure 4.12: Two orders with multiple paths: situation as in Proposition 6, SO solution in cyan.

We will now give the Proposition corresponding to the example network given in Figure 4.10.

Proposition 2 (Case 1 with two orders, $|\mathcal{P}_w| = 2\ \forall\ w \in \mathcal{W}$)**.** *Given is an SO problem with two orders with demand 1 and capacity 1 on all traveling arcs. Assume the paths are ordered such that*

$$C_1^{r_1} \leq C_1^{r_2} \leq \dots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \dots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

where $r_j \in \mathcal{P}_1\ \forall\ j \in \{1, \dots, n\}$ and $p_j \in \mathcal{P}_2\ \forall\ j \in \{1, \dots, m\}$. If the SO solution contains $\underline{f_{r_i}} = 1$ with $C_1^{r_i} = C_1^{r_1}$ and $\underline{f_{q_i}} = 1$ with $C_2^{q_i} = C_2^{q_1}$, then there are no tolls needed because the SO solution is a UE.

Proof. For both orders the cheapest path can be traversed, so for every individual order the path choice is optimal, so a UE is reached. □

In the following proposition both orders cannot travel via their cheapest path, so we need to assign toll to paths in $h_{out,1} \cup h_{out,2}$. We show that it is possible to give a solution with only tolls on paths in those path sets, and thus no tolls added to paths in $h_{in,1} \cup h_{in,2}$.

Proposition 3 (Case 1 with two orders, $|\mathcal{P}_w| \geq 2 \ \forall \ w \in \mathcal{W}$)**.** *Given is an SO problem with two orders with demand 1 and capacity 1 on all traveling arcs. Assume the paths are ordered such that*

$$C_1^{r_1} \leq C_1^{r_2} \leq \cdots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \ \text{ and } \ C_2^{q_1} \leq C_2^{q_2} \leq \cdots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

where $r_j \in \mathcal{P}_1 \ \forall \ j \in \{1, \dots, n\}$ and $p_j \in \mathcal{P}_2 \ \forall \ j \in \{1, \dots, m\}$. If we have an SO solution with $r_k \in h_{in,1}$, $2 \leq k \leq n$ and $q_l \in h_{in,2}$, $2 \leq l \leq m$, it is sufficient to have only tolls on the paths in the path set $\bigcup_{w \in \mathcal{W}} h_{out,w}$ and thus $\beta_w^p = 0 \ \forall \ p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ such that the tolled SO-β is a UE.

Proof. We want to find a solution of the NP-β with $\beta_w^p = 0$ for all $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$.
We have two orders, so $\mathcal{W} = \{1, 2\}$, with $d_1 = d_2 = 1$ and paths for each order:

$$\mathcal{P}_1 = \{r_1, r_2, \dots, r_n\},\ \mathcal{P}_2 = \{q_1, q_2, \dots, q_m\}.$$

So $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 = \{r_1, \dots, r_n, q_1, \dots, q_m\}$. Assume the paths are ordered such that

$$C_1^{r_1} \leq C_1^{r_2} \leq \cdots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \cdots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

and assume the SO solution value is $\sum_{p \in \mathcal{P}} C_1^p \underline{f_p} = C_1^{r_a} + C_2^{q_b}$, with $r_a \neq r_1$ and $q_b \neq q_1$, so s^1 contains $f_{r_a}^1 = 1$, $f_{q_b}^1 = 1$ and $f_p^1 = 0 \ \forall \ p \in \mathcal{P} \setminus \{r_a, q_b\}$. Then

$$C_1^{r_a} + C_2^{q_b} \leq C_1^{r_k} + C_2^{q_l} \ \forall \ s^j = (\boldsymbol{x}^j, \boldsymbol{f}^j) \in \mathcal{S} \setminus \{s^1\} \text{ with } f_{r_k}^j = 1,\ f_{q_l}^j = 1, \text{ and } f_p^j = 0 \ \forall \ p \in \mathcal{P} \setminus \{r_k, q_l\}.$$

Then the SO solution is flow on path r_a and on q_b, thus we have

$$h_{in,1} = \{r_a\},\ h_{in,2} = \{q_b\},\ h_{out,1} = \mathcal{P}_1 \setminus \{r_a\},\ h_{in,2} = \{q_b\},\ h_{out,2} = \mathcal{P}_2 \setminus \{q_b\}$$

Because both orders do not travel via their cheapest paths, tolls on the chosen paths should be negative: $\beta_1^{r_a} \leq 0,\ \beta_2^{q_b} \leq 0$.
The constraint of the NP-β, $\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p = 0$, gives us $\beta_1^{r_a} = -\beta_2^{q_b}$ and with the fact we want the tolls on to used paths to be negate, we obtain $\beta_1^{r_a} = \beta_2^{q_b} = 0$. Then we can derive the following toll constraints on the toll variables:

$$\begin{array}{ccccc}
\beta_1^{r_a} - \beta_1^{r_1} \leq C_1^{r_1} - C_1^{r_a} & \Leftrightarrow & -\beta_1^{r_1} \leq C_1^{r_1} - C_1^{r_a} & \Leftrightarrow & \beta_1^{r_1} \geq C_1^{r_a} - C_1^{r_1} > 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\beta_1^{r_a} - \beta_1^{r_{a-1}} \leq C_1^{r_{a-1}} - C_1^{r_a} & \Leftrightarrow & -\beta_1^{r_{a-1}} \leq C_1^{r_{a-1}} - C_1^{r_a} & \Leftrightarrow & \beta_1^{r_{a-1}} \geq C_1^{r_a} - C_1^{r_{a-1}} > 0 \\
\beta_1^{r_a} - \beta_1^{r_{a+1}} \leq C_1^{r_{a+1}} - C_1^{r_a} & \Leftrightarrow & -\beta_1^{r_{a+1}} \leq C_1^{r_{a+1}} - C_1^{r_a} & \Leftrightarrow & \beta_1^{r_{a+1}} \geq C_1^{r_a} - C_1^{r_{a+1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\beta_1^{r_a} - \beta_1^{r_n} \leq C_1^{r_n} - C_1^{r_a} & \Leftrightarrow & -\beta_1^{r_n} \leq C_1^{r_n} - C_1^{r_a} & \Leftrightarrow & \beta_1^{r_n} \geq C_1^{r_a} - C_1^{r_n} \\
\beta_2^{q_b} - \beta_2^{q_1} \leq C_2^{q_1} - C_2^{q_b} & \Leftrightarrow & -\beta_2^{q_1} \leq C_2^{q_1} - C_2^{q_b} & \Leftrightarrow & \beta_2^{q_1} \geq C_2^{q_b} - C_2^{q_1} > 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\beta_2^{q_b} - \beta_2^{q_{b-1}} \leq C_2^{q_{b-1}} - C_2^{q_b} & \Leftrightarrow & -\beta_2^{q_{b-1}} \leq C_2^{q_{b-1}} - C_2^{q_b} & \Leftrightarrow & \beta_2^{q_{b-1}} \geq C_2^{q_b} - C_2^{q_{b-1}} > 0 \\
\beta_2^{q_b} - \beta_2^{q_{b+1}} \leq C_2^{q_{b+1}} - C_2^{q_b} & \Leftrightarrow & -\beta_2^{q_{b+1}} \leq C_2^{q_{b+1}} - C_2^{q_b} & \Leftrightarrow & \beta_2^{q_{b+1}} \geq C_2^{q_b} - C_2^{q_{b+1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\beta_2^{q_b} - \beta_2^{q_m} \leq C_2^{q_m} - C_2^{q_b} & \Leftrightarrow & -\beta_2^{q_m} \leq C_2^{q_m} - C_2^{q_b} & \Leftrightarrow & \beta_2^{q_m} \geq C_2^{q_b} - C_2^{q_m}
\end{array}$$

Which give us

$$\begin{aligned} \beta_1^{r_1} &= C_1^{r_a} - C_1^{r_1} > 0 \\ &\vdots \\ \beta_1^{r_{a-1}} &= C_1^{r_a} - C_1^{r_{a-1}} > 0 \\ \beta_1^{r_{a+1}} &= 0 \\ &\vdots \\ \beta_1^{r_n} &= 0 \end{aligned} \quad \text{and} \quad \begin{aligned} \beta_2^{q_1} &= C_2^{q_b} - C_2^{q_1} > 0 \\ &\vdots \\ \beta_2^{q_{b-1}} &= C_2^{q_b} - C_2^{q_{b-1}} > 0 \\ \beta_2^{q_{b+1}} &= 0 \\ &\vdots \\ \beta_2^{q_m} &= 0 \end{aligned}$$

Adding these tolls give the following costs for the SO-β problem:

$$\begin{aligned} C_{\beta 1}^{r_1} &= C_1^{r_1} + \beta_1^{r_1} = C_1^{r_a} \\ &\vdots \\ C_{\beta 1}^{r_{a-1}} &= C_1^{r_{a-1}} + \beta_1^{r_{a-1}} = C_1^{r_a} \\ C_{\beta 1}^{r_{a+1}} &= C_1^{r_{a+1}} + \beta_1^{r_{a+1}} = C_1^{r_{a+1}} \\ &\vdots \\ C_{\beta 1}^{r_n} &= C_1^{r_n} + \beta_1^{r_n} = C_1^{r_n} \end{aligned} \quad \text{and} \quad \begin{aligned} C_{\beta 2}^{q_1} &= C_2^{q_1} + \beta_2^{q_1} = C_2^{q_b} \\ &\vdots \\ C_{\beta 2}^{q_{b-1}} &= C_2^{q_{b-1}} + \beta_2^{q_{b-1}} = C_2^{q_b} \\ C_{\beta 2}^{q_{b+1}} &= C_2^{q_{b+1}} + \beta_2^{q_{b+1}} = C_2^{q_{b+1}} \\ &\vdots \\ C_{\beta 2}^{q_m} &= C_2^{q_m} + \beta_2^{q_m} = C_2^{q_m} \end{aligned}$$

so the SO-β solution is a UE, because now for each order, all path costs are higher or equal to the cost of the chosen path. □

We now give results for a situation as given in Figure 4.13.

Proposition 4 (Case 1 with two orders, $|\mathcal{P}_w| = 2 \ \forall \ w \in \mathcal{W}$)**.** *Given is an SO problem with two orders with demand 1 and capacity 1 on all traveling arcs. r_1, r_2 are the paths of order 1, and q_1, q_2 the paths of order 2. If the paths are ordered such that*

$$C_1^{r_1} < C_1^{r_2} \ \text{ and } \ C_2^{q_1} < C_2^{q_2}$$

and the following holds:

$$C_1^{r_1} + C_2^{q_2} \leq C_1^{r_2} + C_2^{q_1},$$

then it is sufficient to have only tolls on the paths in the path set $\bigcup_{w\in\mathcal{W}} h_{in,w}$ and thus $\beta_w^p = 0 \ \forall \ p \in \bigcup_{w\in\mathcal{W}} h_{out,w}$, such that the SO-$\beta$ is a UE.

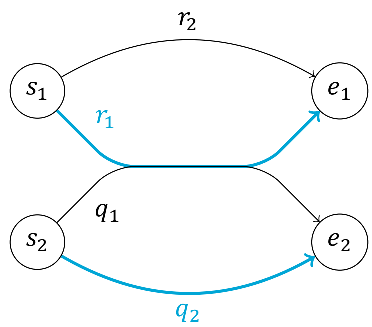


Figure 4.13: SO solution as in Proposition 4

Proof. We want to find a solution with $\beta_w^p = 0$ for all $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$.

We have two path sets: $\mathcal{P}_1 = \{r_1, r_2\}$, $\mathcal{P}_2 = \{q_1, q_2\}$. So $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 = \{r_1, r_2, q_1, q_2\}$. Solving the SO problem gives

$$h_{in,1} = \{r_1\},\ h_{out,1} = \{r_2\},\ h_{in,2} = \{q_2\},\ h_{out,2} = \{q_1\}.$$

In order to reach a UE after applying tolls, we need a negative toll on path q_2: $\beta_2^{q_2} < 0$ (because $C_2^{q_2} > C_2^{q_1}$). The constraint $\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p = 0$ gives us $\beta_1^{r_1} = -\beta_2^{q_2}$ and thus $\beta_1^{r_1} > 0$. We will show a solution exists when we state $\beta_w^p = 0\ \forall\ p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$. We have the following toll constraints:

$$\begin{aligned}
-C_2^{q_2} \leq \beta_2^{q_2} \quad &\wedge \quad \beta_2^{q_2} - \beta_2^{q_1} \leq C_2^{q_1} - C_2^{q_2} < 0 \\
0 \leq \beta_1^{r_1} \quad &\wedge \quad \beta_1^{r_1} - \beta_1^{r_2} \leq C_1^{r_2} - C_1^{r_1} \\
&\Leftrightarrow \\
-C_2^{q_2} \leq \beta_2^{q_2} \quad &\wedge \quad \beta_2^{q_2} \leq C_2^{q_1} - C_2^{q_2} < 0 \\
0 \leq \beta_1^{r_1} \quad &\wedge \quad \beta_1^{r_1} \leq C_1^{r_2} - C_1^{r_1} \\
&\Leftrightarrow
\end{aligned}$$

$$\begin{gathered}
-C_2^{q_2} \leq \beta_2^{q_2} \leq C_2^{q_1} - C_2^{q_2} < 0 \\
0 \leq \beta_1^{r_1} \leq C_1^{r_2} - C_1^{r_1} \\
\Leftrightarrow \\
C_2^{q_2} \geq -\beta_2^{q_2} \geq C_2^{q_2} - C_2^{q_1} > 0 \\
0 \leq \beta_1^{r_1} \leq C_1^{r_1} - C_1^{r_1}
\end{gathered}$$

Using $\beta_1^{r_1} = -\beta_2^{q_2}$ gives us the four inequalities:

$$\begin{aligned}
0 < C_2^{q_2} - C_2^{q_1} \leq \beta_1^{r_1} \leq C_2^{q_2} \\
0 \leq \beta_1^{r_1} \leq C_1^{r_1} - C_1^{r_1}
\end{aligned}
\quad \wedge \quad
\begin{aligned}
-C_2^{q_2} \leq \beta_2^{q_2} \leq C_2^{q_1} - C_2^{q_2} < 0 \\
C_1^{r_1} - C_1^{r_1} \leq \beta_2^{q_2} \leq 0
\end{aligned}$$

Which give us

$$\max\{C_2^{q_2} - C_2^{q_1}, 0\} \leq \beta_1^{r_1} \leq \min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} \tag{4.12}$$

$$\wedge$$

$$\max\{-C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} \leq \beta_2^{q_2} \leq \min\{C_2^{q_1} - C_2^{q_2}, 0\} \tag{4.13}$$

$$\Leftrightarrow$$

$$C_2^{q_2} - C_2^{q_1} \leq \beta_1^{r_1} \leq \min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} \tag{4.14}$$

$$\wedge$$

$$\max\{-C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} \leq \beta_2^{q_2} \leq C_2^{q_1} - C_2^{q_2} \tag{4.15}$$

Note that Equation 4.14 and Equation 4.15 are equivalent, because

$$\beta_1^{r_1} = -\beta_2^{q_2} \quad \text{and} \quad \min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} = -\max\{-C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\}.$$

So a solution exists when

$$C_2^{q_2} - C_2^{q_1} \leq \beta_1^{r_1} \leq \min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\}. \tag{4.16}$$

This inequality is always valid (given that $C_1^{r_1} + C_2^{q_2} \leq C_1^{r_2} + C_2^{q_1}$ as assumed):

- If $\min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} = C_2^{q_2}$, then (4.16): $C_2^{q_2} - C_2^{q_1} < C_2^{q_2} = \min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\}$ is valid.

- If $\min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\} = C_1^{r_1} - C_1^{r_1}$, then (4.16): $C_2^{q_2} - C_2^{q_1} \leq C_1^{r_1} - C_1^{r_1} = \min\{C_2^{q_2}, C_1^{r_1} - C_1^{r_1}\}$ is valid because of the assumptions: $C_1^{r_1} + C_2^{q_2} \leq C_1^{r_2} + C_2^{q_1} \Leftrightarrow C_2^{q_2} - C_2^{q_1} \leq C_1^{r_2} - C_1^{r_1}$.

So we can always obtain a solution for the NP-β such that

$$C_{\beta 1}^{r_1} = C_1^{r_1} + \beta_1^{r_1} < C_1^{r_1} = C_{\beta 1}^{r_2},$$
$$C_{\beta 2}^{q_2} = C_2^{q_2} + \beta_2^{q_2} < C_2^{q_1} = C_{\beta 2}^{q_1},$$

so the SO-β solution is a UE. □

We will now give results for situations as given in Figure 4.11

Proposition 5 (Case 1 with two orders, $|\mathcal{P}_w| \geq 2 \ \forall \ w \in \mathcal{W}$)**.** *Given is an SO problem with two orders with demand 1 and capacity 1 on all traveling arcs. Assume the paths are ordered such that*

$$C_1^{r_1} \leq C_1^{r_2} \leq \cdots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \cdots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

where $r_j \in \mathcal{P}_1 \ \forall \ j \in \{1, \ldots, n\}$ and $p_j \in \mathcal{P}_2 \ \forall \ j \in \{1, \ldots, m\}$. If the SO solution s^1 contains the cheapest path for one of the orders, and a non-optimal path for the other order, there are two options:

(i) Assuming paths r_1, q_l, $2 \leq l \leq m$ are in the SO solution and $C_2^{q_l} - C_2^{q_1} \leq C_1^{r_2} - C_1^{r_1}$, or assuming paths q_1, r_k, $2 \leq k \leq n$ are in the SO solution and $C_1^{r_k} - C_1^{r_1} \leq C_2^{q_2} - C_2^{q_1}$, then all tolls for paths $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$ are zero, tolls can be added to paths $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$, so $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$,

(ii) Assuming paths r_1, q_l, $2 \leq l \leq m$ are in the SO solution and $C_2^{q_l} - C_2^{q_1} \geq C_1^{r_2} - C_1^{r_1}$, or assuming paths q_1, r_k, $2 \leq k \leq n$ are in the SO solution and $C_1^{r_k} - C_1^{r_1} \geq C_2^{q_2} - C_2^{q_1}$, then all tolls for paths $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ are zero, tolls can be added to paths $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$, so $\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} |\beta_w^p| = 0$.

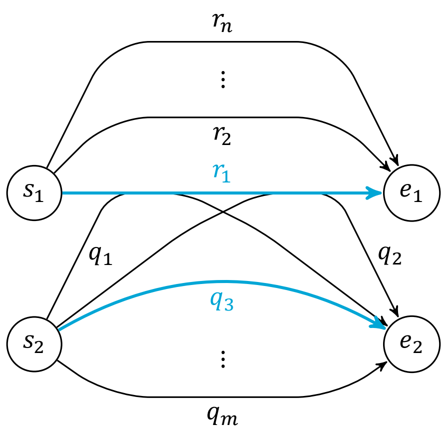


Figure 4.14: An example of a situation as in Proposition 5 with $q_l = q_3$.

Proof. (i) In this case (assume r_1, q_l in SO solution, solving the case where r_k, q_1 are in the SO solution can be solved in the same way), we want the toll of the non-optimal chosen path (here that is q_l) to be negative, such that the adjusted path cost $C_{\beta 2}^{q_l}$ is equal to the initial cheapest path cost $k_2 = C_2^{q_1}$, thus $C_{\beta 2}^{q_l} = C_2^{q_1}$. The adjusted path cost is $C_{\beta 2}^{q_l} = C_2^{q_l} + \beta_2^{q_l}$. So we want

$$\beta_2^{q_l} = C_{\beta 2}^{q_l} - C_2^{q_l} = C_2^{q_1} - C_2^{q_l},$$

which is negative because $C_2^{q_1} \leq C_2^{q_l}$.
Because we only have two paths in the SO solution, we need $-\beta_2^{q_l} = \beta_1^{r_1}$ in order to obtain

$\sum_{w\in\mathcal{W}} \sum_{p\in h_{in,w}} \beta_w^p = 0$. For $\beta_1^{r_1}$ we have the restriction $\beta_1^{r_1} \leq C_1^{r_2} - C_1^{r_1}$ in order to avoid that the cheapest path r_1 becomes more expensive than its last but one cheapest path r_2. Then we obtain the restriction

$$-\left(C_2^{q_1} - C_2^{q_l}\right) = -\beta_2^{q_l} = \beta_1^{r_1} \leq C_1^{r_2} - C_1^{r_1}.$$

So in order to have a valid solution, the following inequality must hold:

$$C_2^{q_l} - C_2^{q_1} \leq C_1^{r_2} - C_1^{r_1},$$

which is what we assumed. So it is sufficient to only add tolls to the paths $p \in \bigcup_{w\in\mathcal{W}} h_{in,w}$.

(ii) In this case (assume r_1, q_l in SO solution, solving the case where r_k, q_1 are in the SO solution can be solved in the same way), suppose we only add tolls to the paths $p \in \bigcup_{w\in\mathcal{W}} h_{in,w}$. Then $-\beta_2^{q_l} > \beta_1^{r_1}$, which does not match the constraint $\sum_{w\in\mathcal{W}} \sum_{p\in h_{in,w}} \beta_w^p = 0$. So we have two options left: or adding only tolls on paths $p \in h_{out,w}$, or adding tolls on both paths in $h_{out,w}$ as well in $h_{in,w}$. The tolls obtained by solving the NP-β suffice the fact that tolls on the unused paths $p \in \bigcup_{w\in\mathcal{W}} h_{out,w}$ are minimized and $\sum_{w\in\mathcal{W}} \sum_{p\in h_{in,w}} \beta_w^p = 0$. This implies that the toll solution will add tolls also on paths $p \in h_{in,w}$, because then the objective value $\sum_{w\in\mathcal{W}} \sum_{p\in h_{out,w}} |\beta_w^p|$ is minimal.

□

We show an example of Proposition 5.(ii):

Example 14. We use the network given in Figure 4.15. The path costs in this network are $C_1^{r_1} = 1$, $C_1^{r_2} = 4$, $C_1^{r_3} = 5$, $C_2^{q_1} = 5$, $C_2^{q_2} = 6$, $C_2^{q_3} = 7$. The SO solution s^1 as denoted in Figure 4.15 has total cost $1 + 7 = 8$. There are two toll solutions:

1. Tolls on paths in $h_{in,w}$ as well on paths in $h_{out,w}$: $\beta_1^{r_1} = 1$, $\beta_2^{q_3} = -1$, $\beta_2^{q_1} = 1 \Rightarrow C_{\beta 1}^{r_1} = 2$, $C_{\beta 2}^{q_3} = 6$, $C_{\beta 2}^{q_1} = 6$,

2. Tolls only on paths in $h_{out,w}$: $\beta_{q_1} = 2$, $\beta_{q_2} = 1 \Rightarrow C_{\beta 2}^{q_1} = 7$, $C_{\beta 2}^{q_2} = 7$.

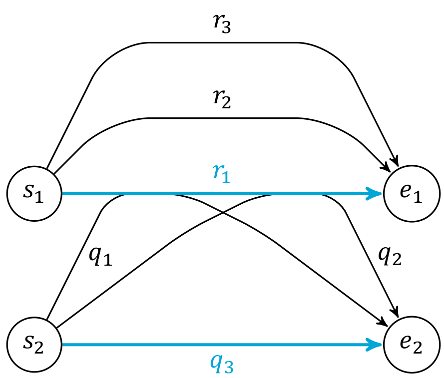


Figure 4.15: An example of a situation as in Proposition 5 with $C_1^{r_1} = 1$, $C_1^{r_2} = 4$, $C_1^{r_3} = 5$, $C_2^{q_1} = 5$, $C_2^{q_2} = 6$, $C_2^{q_3} = 7$.

In both solutions the total absolute value of all tolls is 3, but $\sum_{w\in\mathcal{W}}\sum_{p\in h_{out,w}} |\beta_w^p| = 1$ for solution 1 and $\sum_{w\in\mathcal{W}}\sum_{p\in h_{out,w}} |\beta_w^p| = 3$ for solution 2. So if we want to minimize the NP-β objective, the best toll solution will be adding tolls on both the chosen paths as on the unchosen paths, so solution 1. $\triangle$

Case 1 with multiple orders

In the next proposition we will show what conditions need to hold in order to obtain path tolls with $\beta_w^p = 0 \ \forall\ p \in \bigcup h_{out,w}$.

Proposition 6 (Case 1 with for $n \geq 2$ orders)**.** *Given is an SO solution in Case 1 with orders $w \in \mathcal{W}$, define path sets $h_{in,w}$ and $h_{out,w}$ $\forall\ w \in \mathcal{W}$. Define $\mathcal{P}_w = \{p_1, \dots, p_n\}$ with $C_w^{p_1} \leq \dots \leq C_w^{p_n}$ $\forall\ w \in \mathcal{W}$. Define paths $p_i \in h_{in,w}$, $p_j \in h_{out,w}$. If $C_w^{p_i} \geq C_w^{p_j}$ for one or more orders $w \in \mathcal{W}$, it is sufficient to have only tolls on the paths in the path set $\bigcup_{w\in\mathcal{W}} h_{in,w}$ and thus $\beta_p = 0\ \forall\ p \in \bigcup_{w\in\mathcal{W}} h_{out,w}$ such that the SO-β is a UE $\Leftrightarrow$*

$$\sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}} \left(C_w^{p_i} - C_w^{p_1}\right) \leq \sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i= p_1}} \left(C_w^{p_2} - C_w^{p_1}\right)$$
$$\Leftrightarrow \sum_{w\in\mathcal{W}}\sum_{p_i\in h_{in,w}} C_w^{p_i} \leq \sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}} C_w^{p_1} + \sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i= p_1}} C_w^{p_2},$$

with $p_i \in h_{in,w}\ \forall\ w \in \mathcal{W}$.

Proof. We want to find a solution with $\beta_p = 0$ for all $p \in \bigcup_{w\in\mathcal{W}} h_{out,w}$.

Define $\mathcal{W} = \{1, 2, \dots, n-1, n\}$ and $d_w = 1\ \forall\ w \in \mathcal{W}$ and path sets for each order: $\mathcal{P}_w\ \forall\ w \in \mathcal{W}$. Assume the paths are ordered such that

$$C_w^{p_1} \leq C_w^{p_2} \leq \dots \leq C_w^{p_{m-1}} \leq C_w^{p_m},\ \forall\ w \in \mathcal{W} \text{ with } m = |\mathcal{P}_w|,$$

so $\mathcal{P}_w = \{p_1, \dots, p_m\}$ $\forall\ w \in \mathcal{W}$, and assume there exists an SO solution: for each order w denote the chosen path by p_i. Then

$$h_{in,w} = \{p_i\},\ h_{out,w} = \mathcal{P}_w \setminus \{p_i\}\ \forall\ w \in \mathcal{W}.$$

We can say that if other solutions exist (so if $|\mathcal{S}| > 2$), the SO cost satisfies the following inequality:

$$C_1^{p_i} + C_2^{p_i} + \dots + C_{n-1}^{p_i} + C_n^{p_i} \leq C_1^{p_j} + C_2^{p_j} + \dots + C_{n-1}^{p_j} + C_n^{p_j}$$

with $p_i \in h_{in,w}\ \forall\ w \in \mathcal{W}$ and p_j the chosen paths in solution s^j.

If the UE solution does not equal the SO solution (so there are tolls needed to obtain a UE when traveling via the paths chosen in the SO solution), then there are paths chosen in the SO for which the path cost is not equal to the cheapest path cost (otherwise the SO solution is already equal to the UE solution). So for at least one order w it holds that $C_w^{p_i} > C_w^{p_j}$ for some $p_j \in h_{out,w}$, $p_i \in h_{in,w}$.

We add some extra notation: for each path in the SO solution for each order we define the path sets

$$g_w^{p_i} = \{p_1, \dots, p_{i-1}\},\quad l_w^{p_i} = \{p_{i+1}, \dots, p_m\}.$$

All paths in $g_w^{p_i}$ have lower cost than paths p_i and all paths in $l_w^{p_i}$ have higher cost. So in Case 1 in which we are now working (recall solutions of Case 1 have one chosen path per order), we can split $h_{out,w}$ into those two path sets for each order $w \in \mathcal{W}$:

$$h_{out,w} = g_w^{p_i} \cup l_w^{p_i}\ \forall\ w \in \mathcal{W}.$$

If $p_i \neq p_1$, we know $p_i \in g_w^{p_i} \neq \emptyset$, and we require $\beta_w^{p_i} < 0$. Since we only require tolls on paths in $\bigcup_{w \in \mathcal{W}} h_{in,w}$, we can state

$$\sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} \beta_w^{p_i} = - \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \beta_w^{p_1}$$

Then we obtain the following bounds for the tolls:

$-C_w^{p_i} \leq \beta_w^{p_i} \leq C_w^{p_j} - C_w^{p_i} < 0 \ \forall \, p_j \in g_w^{p_i} \neq \emptyset, \ \ w \in \mathcal{W}$

$-C_w^{p_i} \leq \beta_w^{p_i} \leq C_w^{p_h} - C_w^{p_i} \ \forall \, p_h \in l_w^{p_i}, \ g_w^{p_i} \neq \emptyset, \ w \in \mathcal{W}$ (redundant because $C_w^{p_j} - C_w^{p_i} < 0 < C_w^{p_h} - C_w^{p_i}$)

$0 \leq \beta_w^{p_1} \leq C_w^{p_h} - C_w^{p_1} \ \forall \, p_h \in l_w^{p_i}, \ g_w^{p_i} = \emptyset, \ w \in \mathcal{W}$

$\Leftrightarrow$

$-C_w^{p_i} \leq \beta_w^{p_i} \leq \min_{1 \leq j \leq i-1} \left\{ C_w^{p_1} - C_w^{p_i}, \dots, C_w^{p_{i-1}} - C_w^{p_i} \right\} < 0, \ g_w^{p_i} \neq \emptyset, \ w \in \mathcal{W}$

$0 \leq \beta_w^{p_1} \leq \min_{2 \leq h \leq m} \left\{ C_w^{p_2} - C_w^{p_1}, \dots, C_w^{p_m} - C_w^{p_1} \right\}, \ g_w^{p_i} = \emptyset, \ w \in \mathcal{W}$

$\Leftrightarrow$

$-C_w^{p_i} \leq \beta_w^{p_i} \leq C_w^{p_1} - C_w^{p_i} < 0, \ g_w^{p_i} \neq \emptyset, \ w \in \mathcal{W}$

$0 \leq \beta_w^{p_1} \leq C_w^{p_2} - C_w^{p_1}, \ g_w^{p_i} = \emptyset, \ w \in \mathcal{W}$

Then when we add all constraints with $g_w^{p_i} \neq \emptyset$ (so $p_i \neq p_1$) and add all constraints with $g_w^{p_i} = \emptyset$ (so $p_i = p_1$), we obtain

$$\sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} -C_w^{p_i} \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} \beta_w^{p_i} \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} \left(C_w^{p_1} - C_w^{p_i} \right)$$

$$0 \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \beta_w^{p_1} \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \left(C_w^{p_2} - C_w^{p_1} \right)$$

$$\Leftrightarrow$$

$$\sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} C_w^{p_i} \geq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} -\beta_w^{p_i} \geq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} \left(C_w^{p_i} - C_w^{p_1} \right)$$

$$0 \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \beta_w^{p_1} \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \left(C_w^{p_2} - C_w^{p_1} \right)$$

$$\Leftrightarrow$$

$$\max \left\{ \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} \left(C_w^{p_i} - C_w^{p_1} \right), 0 \right\} \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \beta_w^{p_1} = \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} -\beta_w^{p_i}$$

$$\leq \min \left\{ \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} C_w^{p_i}, \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \left(C_w^{p_2} - C_w^{p_1} \right) \right\}$$

$$\Leftrightarrow$$

$$\sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} \left(C_w^{p_i} - C_w^{p_1} \right) \leq \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \beta_w^{p_1} = \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} -\beta_w^{p_i} \leq \min \left\{ \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i \neq p_1}} C_w^{p_i}, \sum_{w \in \mathcal{W}} \sum_{\substack{p_i \in h_{in,w} \\ p_i = p_1}} \left(C_w^{p_2} - C_w^{p_1} \right) \right\}$$

This inequality is only valid if

$$\sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}}\left(C_w^{p_i}-C_w^{p_1}\right)\leq \min\left\{\sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}}C_w^{p_i},\sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i=p_1}}\left(C_w^{p_2}-C_w^{p_1}\right)\right\}$$

holds. We can easily verify that

$$\sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}}\left(C_w^{p_i}-C_w^{p_1}\right)\leq \sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}}C_w^{p_i}$$

is always valid. Then we need to make verify the other inequality:

$$\sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i\neq p_1}}\left(C_w^{p_i}-C_w^{p_1}\right)\leq \sum_{w\in\mathcal{W}}\sum_{\substack{p_i\in h_{in,w}\\ p_i=p_1}}\left(C_w^{p_2}-C_w^{p_1}\right).$$

So if this inequality holds, then there are no tolls necessary on paths $p\in h_{out,w}\ \forall\ w\in\mathcal{W}$. □

We showed how tolls in Case 1 behave, by expanding the problem in order size. We started with problems with one order, and expanding the order size to n orders. When adding tolls, we aim for a solution in which as little tolls as possible are added to paths which are not in the SO solution, so paths in $\bigcup_{w\in\mathcal{W}} h_{out,w}$. So the main result of this section is Proposition 6, which states what conditions on the path costs are needed in order to obtain a toll solution with no tolls added to paths $p\in\bigcup_{w\in\mathcal{W}} h_{out,w}$.

4.4. Case 2

Case 2 is the situation in which $d_w\geq 1\ \forall\ w\in\mathcal{W}$, and $m_a=1\ \forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a=\infty$.
We show an example of solving a Case 2 problem in Subsection 4.4.1 and in Subsection 4.4.2 we will derive properties for path tolls.

4.4.1. Case 2: Example

Example 15. We use the STN in Figure 4.16 with two orders of size 2: $d_1=d_2=2$.
The possible paths for the orders are:

$$\mathcal{P}_1=\{a,bc,e,f\},\ \mathcal{P}_2=\{b,ad,ag,eg,fg\}.$$

The SO solution is given in Figure 4.16, with s_w and e_w denoting the end point of order w, respectively.

We have path costs

$$C_1^a=2,\ C_1^e=4,\ C_1^{bc}=4,\ C_1^f=5,\ C_2^b=3,\ C_2^{ad}=4,\ C_2^{ag}=6,\ C_2^{eg}=6,\text{ and } C_2^{fg}=6,$$

and the path sets following from the SO solution:

$$h_{in,1}=\{a,e\},\ h_{in,2}=\{b,fg\},\ h_{out,1}=\{f\},\ \text{ and } h_{out,2}=\{ad,ag,eg\}.$$

The order costs are

$$C_1=C_1^a+C_1^e=2+4=6,\qquad C_2=C_2^b+C_2^{fg}=3+6=9.$$

We can easily see that $k_1=C_1^a+C_1^e=2+4=6$ and $k_2=C_2^b+C_2^{ad}=3+4=7$, so in this case order 2 does not reach its optimum. We thus do not have a UE and we need to assign tolls to the STN in order

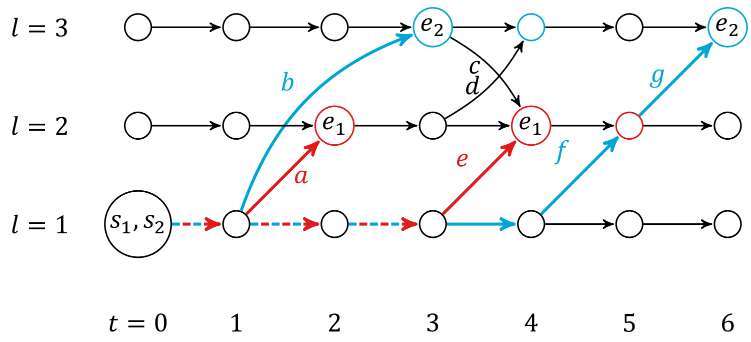


Figure 4.16: STN with two orders, $d_w = 2 \; \forall \; w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between different locations, $m_{a_i} = \infty$ on waiting arcs. The denoted solution is SO.

to reach a UE.

We can see that for both orders, three or more paths are possible, from which two paths have to be traversed. For order 1 there are paths of costs 2, 4 and 5 and for order 2 there are paths of costs 3, 4 and 6 (there are multiple paths with cost 6). To calculate the tolls, we again use the NP-β in Algorithm 2:

$$\min \left|\beta_1^f\right| + \left|\beta_1^{bc}\right| + \left|\beta_2^{ad}\right| + \left|\beta_2^{ag}\right| + \left|\beta_2^{eg}\right|$$

$$\text{Subject to } \beta_1^a + \beta_1^e + \beta_2^b + \beta_2^{fg} = 0 \tag{4.17}$$

$$\begin{aligned} \beta_1^a - \beta_1^f \leq 3, \quad & \beta_1^a - \beta_1^{bc} \leq 2, \\ \beta_1^e - \beta_1^f \leq 1, \quad & \beta_1^e - \beta_1^{bc} \leq 0, \\ \beta_2^b - \beta_2^{ad} \leq 1, \quad \beta_2^b - \beta_2^{ag} \leq 3, \quad & \beta_2^b - \beta_2^{eg} \leq 3, \\ \beta_2^{fg} - \beta_2^{ad} \leq -2, \quad \beta_2^{fg} - \beta_2^{ag} \leq 0, \quad & \beta_2^{fg} - \beta_2^{eg} \leq 0 \end{aligned} \tag{4.18}$$

$$\begin{aligned} \beta_1^a \geq -2, \quad \beta_1^{bc} \geq -4, \quad \beta_1^e \geq -4, \quad & \beta_1^f \geq -5, \\ \beta_2^b \geq -3, \quad \beta_2^{ad} \geq -4, \quad \beta_2^{ag} \geq -6, \quad \beta_2^{eg} \geq -6, \quad & \beta_2^{fg} \geq -6. \end{aligned} \tag{4.19}$$

A valid solution is $\beta_1^a = 2$, $\beta_2^{fg} = -2$ and 0 for all other path tolls. So the objective value of this NP-β is 0, which means no toll on paths $p \in \bigcup_{w \in \mathcal{W}} h_{out,w}$ are assigned. These tolls give us the new path costs

$$\begin{aligned} C_{\beta 1}^a = 2 + 2 = 4, \quad C_{\beta 1}^e = 4, \quad C_{\beta 2}^b = 3, \quad & C_{\beta 2}^{fg} = 6 - 2 = 4, \\ C_{\beta 1}^f = 5, \quad C_{\beta 1}^{bc} = 4, \quad C_{\beta 2}^{ad} = 4, \quad & C_{\beta 2}^{ag} = 6, \quad C_{\beta 2}^{eg} = 6. \end{aligned}$$

So we see that an optimal solution for the SO-β is the solution of the initial SO problem, because all path costs are equal to or higher than the costs of that initial SO solution. $\triangle$

4.4.2. Case 2: Finding properties of path tolls

Case 2 with two orders

For the following propositions, we will define the following: Define $\mathcal{W} = \{1, 2\}$, $d_1 = d_2 = 2$, and paths for each order:

$$\mathcal{P}_1 = \{r_1, r_2, \ldots, r_n\}, \; \mathcal{P}_2 = \{q_1, q_2, \ldots, q_m\}.$$

So $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 = \{r_1, \ldots, r_n, q_1, \ldots, q_m\}$. Assume the paths are ordered such that

$$C_1^{r_1} \leq C_1^{r_2} \leq \cdots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \cdots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

and assume the SO solution value is $C_1^{r_a} + C_1^{r_b} + C_2^{q_c} + C_2^{q_d}$, so the path flow vector $\boldsymbol{f}^1$ of s^1 contains elements $f_{r_a}^1 = 1$, $f_{r_b}^1 = 1$, $f_{q_c}^1 = 1$, $f_{q_d}^1 = 1$ and $f_p^1 = 0 \; \forall \; p \in \mathcal{P} \setminus \{r_a, r_b, q_c, q_d\}$. Then

$$C_1^{r_a} + C_1^{r_b} + C_2^{q_c} + C_2^{q_d} \leq C_1^{r_k} + C_1^{r_l} + C_2^{q_k} + C_2^{q_l} \; \forall \; s^j = \left(\underline{x}^j, \underline{f}^j\right) \in S \setminus \{s^1\}$$
$$\text{with } f_{r_k}^j = 1, \; f_{r_l}^j = 1, \; f_{q_k}^j = 1, \; f_{q_l}^j = 1 \text{ and } f_p^j = 0 \; \forall \; p \in \mathcal{P} \setminus \{r_k, r_l, q_k, q_l\}.$$

Which solutions on the tolls do we obtain?

- If for order 1: $r_1, r_2 \in h_{in,1}$ and for order 2: $q_1, q_2 \in h_{in,2}$, then no tolls are added: $\beta_w^p = 0 \ \forall \ p \in \mathcal{P}$, see Proposition 7 and Figure 4.17.

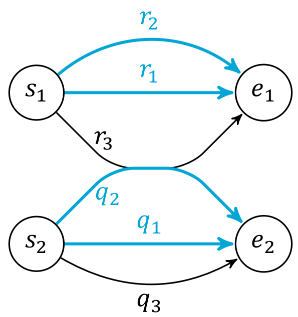


Figure 4.17: Two orders with multiple paths: situation as in Proposition 7, SO solution in cyan.

- If for order 1: $h_{in,1} = \{r_1, r_k\}$, $2 \le k \le n$ and for order 2: $h_{in,2} = \{q_k, q_l\}$, $2 \le k < l \le m$ with the condition

$$C_1^{r_1} + C_1^{r_k} + C_2^{q_k} + C_2^{q_l} \le C_1^{r_k} + C_2^{r_l} + C_2^{q_1} + C_2^{q_k} \iff C_1^{r_1} + C_2^{q_l} \le C_1^{r_l} + C_2^{q_1}$$

  òr
  if for order 1: $h_{in,1} = \{r_k, r_l\}$, $2 \le k < l \le n$ and for order 2: $h_{in,2} = \{q_1, q_k\}$, $2 \le k \le m$ with the condition

$$C_1^{r_k} + C_1^{r_l} + C_2^{q_1} + C_2^{q_k} \le C_1^{r_1} + C_1^{r_k} + C_2^{q_k} + C_2^{q_l} \iff C_1^{r_l} + C_2^{q_1} \le C_1^{r_1} + C_2^{q_l},$$

  then it is sufficient to add tolls on the used paths only, see Proposition 8 and Figure 4.18a and 4.18b (here $C_2^{q_l} - C_2^{q_1} \le C_1^{r_2} - C_1^{r_1}$ with $l = 2$ always holds in Figure 4.18a and $C_1^{r_k} - C_1^{r_1} \le C_2^{q_2} - C_2^{q_1}$ with $k = 2$ always holds in Figure 4.18b).

(a) SO solution in cyan with $C_1^{r_1} + C_1^{r_2} + C_2^{q_2} + C_2^{q_3} \le C_1^{r_2} + C_1^{r_3} + C_2^{q_1} + C_2^{q_2}$.

(b) SO solution in cyan with $C_1^{r_1} + C_1^{r_2} + C_2^{q_2} + C_2^{q_3} \ge C_1^{r_2} + C_1^{r_3} + C_2^{q_1} + C_2^{q_2}$.

Figure 4.18: Two orders with multiple paths: situations as in Proposition 8.

  Note that for Figure 4.19 there is only one solution possible, due to the capacity constraints.

- If for order 1: $h_{in,1} = \{r_k, r_l\}$, $2 \le k < l \le n$ and for order 2: $h_{in,2} = \{q_k, q_l\}$, $2 \le k < l \le m$ then tolls on the unused paths with lower path cost are necessary, see Proposition 8 and Figure 4.20.

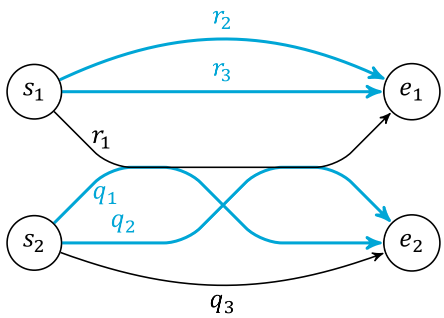


Figure 4.19: SO solution in cyan with
$C_1^{r_2} + C_1^{r_3} + C_2^{q_1} + C_2^{q_2}$ (the only possible solution).

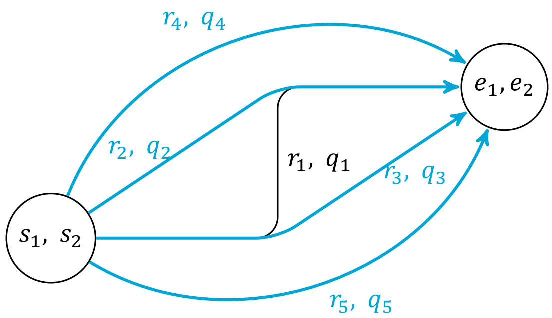


Figure 4.20: Two orders with multiple paths: situation as in Proposition 8, SO solution in cyan.

The following proposition corresponds to Figure 4.17.

Proposition 7 (Case 2 with two orders)**.** *Given is an SO problem with two orders with demand* $d_w > 1$ *for both orders and capacity 1 on all traveling arcs. Assume the paths are ordered such that*

$$C_1^{r_1} \leq C_1^{r_2} \leq \cdots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \cdots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

where $r_j \in \mathcal{P}_1\ \forall\ j \in \{1, \dots, n\}$ *and* $p_j \in \mathcal{P}_2\ \forall\ j \in \{1, \dots, m\}$*. If the SO solution contains* $f_{r_i} = 1\ \forall\ 1 \leq i \leq d_1$ *and* $f_{q_i} = 1\ \forall\ 1 \leq i \leq d_2$*, then there are no tolls needed because the SO solution is a UE.*

Proof. For both orders the cheapest paths can be traversed, so for every individual order the path choice is optimal, so a UE is reached. □

Case 2 with multiple orders

The following proposition gives us the condition which has to hold when we only want tolls on the chosen paths, given an SO solution.

Proposition 8 (Case 2 with multiple orders)**.** *Given is an SO solution with orders* $w \in \mathcal{W}$*, define path sets* $h_{in,w}$ *and* $h_{out,w}$ $\forall\ w \in \mathcal{W}$*. Define* $\mathcal{P}_w = \{p_1, \dots, p_n\}$ *with* $C_w^{p_1} \leq \dots \leq C_w^{p_n}$ $\forall\ w \in \mathcal{W}$*. Define for each path* $p_j \in h_{in,w}$ $\forall\ w \in \mathcal{W}$*:* $g_w^{p_j} := \{p_i \mid i < j,\ p_i \in h_{out,w}\}$, $l_w^{p_j} := \{p_i \mid i > j,\ p_i \in h_{out,w}\}$*. Then to obtain a toll solution where there are only tolls on chosen paths, so* $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$*, the following inequality needs to hold:*

$$\sum_{w \in \mathcal{W}} \sum_{q \in h_{in,w}} C_w^q \leq \sum_{w \in \mathcal{W}} \left(\sum_{q \in h_{in,w}} \min_{r \in h_{out,w}} C_w^r \right).$$

Proof. Assume $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$. The toll constraints are

$$\begin{aligned} \beta_w^q &\geq -C_w^q \ \forall\, q \in h_{in,w} \ \forall\, w \in \mathcal{W} \\ \beta_w^q - \beta_w^r &\leq C_w^r - C_w^q \ \forall\, q \in h_{in,w},\ r \in h_{out,w} \ \forall\, w \in \mathcal{W} \end{aligned}$$

If there is no toll on $h_{out,w}$, then $\beta_w^r = 0 \ \forall\, r \in h_{out,w} \ \forall\, w \in \mathcal{W}$ and thus

$$\begin{gathered} \beta_w^q \geq -C_w^q \ \forall\, q \in h_{in,w} \ \forall\, w \in \mathcal{W} \\ \beta_w^q \leq C_w^r - C_w^q \ \forall\, q \in h_{in,w},\ r \in h_{out,w} \ \forall\, w \in \mathcal{W} \\ \Leftrightarrow \\ -C_w^q \leq \beta_w^q \leq C_w^r - C_w^q \ \forall\, q \in h_{in,w},\ r \in h_{out,w} \ \forall\, w \in \mathcal{W}. \end{gathered}$$

We defined for each path $p_j \in h_{in,w} \ \forall\, w \in \mathcal{W}$ the path sets $g_w^{p_j} := \{p_i \mid i < j,\ p_i \in h_{out,w}\}$ and $l_w^{p_j} := \{p_i \mid i > j,\ p_i \in h_{out,w}\}$. Then for each path $q \in h_{in,w}$ we obtain the following constraints:

$$\begin{aligned} \beta_w^q &\geq -C_w^q \ \forall\, q \in h_{in,w} \ \forall\, w \in \mathcal{W} \\ \beta_w^q &\leq \min_{r \in h_{out,w}} \{C_w^r - C_w^q\} \ \forall\, q \in h_{in,w} \ \forall\, w \in \mathcal{W}. \end{aligned} \tag{4.20}$$

Note that $h_{out,w} = \bigcup_{q \in h_{in,w}} g_w^q \cup l_w^q$, and with this we can rewrite Constraint 4.20:

$$\begin{aligned} \beta_w^q &\leq \min_{r \in g_w^q} \{C_w^r - C_w^q\} < 0 \ \forall\, q \in h_{in,w} \ \forall\, w \in \mathcal{W} \\ \beta_w^q &\leq \min_{r \in l_w^q} \{C_w^r - C_w^q\} \ \forall\, q \in h_{in,w} \ \forall\, g_w = \emptyset,\ w \in \mathcal{W}. \end{aligned} \tag{4.21}$$

If tolls are necessary, we need both positive and negative tolls, because $\sum_{w \in \mathcal{W}} \sum_{q \in h_{in,w}} \beta_w^q = 0$. In order to obtain this zero sum and the fact that we want tolls on chosen paths only, the following needs to hold:

$$\begin{gathered} \sum_{w \in \mathcal{W}} \sum_{\substack{q \in h_{in,w} \\ g_w^q \neq \emptyset}} \beta_w^q = \sum_{w \in \mathcal{W}} \sum_{\substack{q \in h_{in,w} \\ g_w^q = \emptyset}} -\beta_w^q \\ \Leftrightarrow \\ \sum_{w \in \mathcal{W}} \sum_{\substack{q \in h_{in,w} \\ g_w^q \neq \emptyset}} |\beta_w^q| = \sum_{w \in \mathcal{W}} \sum_{\substack{q \in h_{in,w} \\ g_w^q = \emptyset}} |\beta_w^q|. \end{gathered}$$

We know for paths with $g_w^q \neq \emptyset$ that $\beta_w^q < 0$, because if we do not require this, there will be unused paths with cheaper path cost than some used paths. So we can rewrite the equality to

$$-\sum_{w \in \mathcal{W}} \sum_{\substack{q \in h_{in,w} \\ g_w^q \neq \emptyset}} \beta_w^q = \sum_{w \in \mathcal{W}} \sum_{\substack{q \in h_{in,w} \\ g_w^q = \emptyset}} \beta_w^q.$$

This only holds if the inequalities (4.21) are satisfied when we sum over these inequalities:

$$\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\min_{r\in g_w^q}\left(C_w^q-C_w^r\right)\leq-\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\beta_w^q=\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\beta_w^q\leq\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\min_{r\in l_w^q}\left(C_w^r-C_w^q\right)$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\min_{r\in g_w^q}\left(C_w^q-C_w^r\right)\leq\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\min_{r\in l_w^q}\left(C_w^r-C_w^q\right).$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{q\in h_{in,w}}C_w^q\leq\sum_{w\in\mathcal{W}}\left(\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\min_{r\in g_w^q}C_w^r+\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\min_{r\in l_w^q}C_w^r\right)$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{q\in h_{in,w}}C_w^q\leq\sum_{w\in\mathcal{W}}\left(\sum_{q\in h_{in,w}}\min_{r\in h_{out,w}}C_w^r\right)$$

So in order to obtain a tolled UE solution with no tolls on unchosen paths, the last inequality needs to hold. □

As we did for Case 1, we explored the toll solutions for Case 2 based on the initial path costs, and in Proposition 8 we see what conditions are necessary to add tolls on paths used in the SO solution only, which is equivalent to optimally minimize the NP-β.

4.5. Case 3

Case 3 is the situation in which $d_w = 1\ \forall\ w\in\mathcal{W}$, and $m_a \geq 1\ \forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
We show an example of solving a Case 3 problem in Subsection 4.5.1 and in Subsection 4.5.2 we will derive properties for path tolls.

For both Case 3 and Case 4, we have a new definition of path sets and $h_{in,w}$ and $h_{out,w}$. We first define the path capacity as $m_p = \sum_{a\in\mathcal{A}}\delta_{ap}m_a$.
We now define the path sets $\mathcal{P}_{cap,w}$, $h_{in,w}$ and $h_{out,w}$ $\forall\ w\in\mathcal{W}$:

$$\mathcal{P}_{cap,w}=\left\{\begin{matrix}p\\ m_p\end{matrix}\mid p\in\mathcal{P}_w\right\},$$

$$h_{in,w}=\left\{\begin{matrix}p\\ \underline{f_p}\end{matrix}\mid \underline{f_p}>0,\ p\in\mathcal{P}_w\right\},$$

$$h_{out,w}=\left\{\begin{matrix}p\\ m_p-\underline{f_p}\end{matrix}\mid m_p>\underline{f_p},\ p\in\mathcal{P}_w\right\}.$$

4.5.1. Case 3: Example

Example 16. In this example we have 5 orders all with demand 1: Order 1,2, 3 and for start at location 1, order 5 starts at location 2. Order 1 and 2 end at location 2 and the others at location 3. The SO solution is given in Figure 4.21, with s_w and e_w denoting the start end point of order w, respectively. All traveling arcs have capacity 1, except for arcs b and f, which have capacity $m_b = m_f = 2$.

So we get the following path sets per order with corresponding capacities:

$$\mathcal{P}_{cap,1} = \mathcal{P}_{cap,2} = \begin{Bmatrix} a, & b, & c, & d \\ 1, & 2, & 1, & 1 \end{Bmatrix}$$
$$\mathcal{P}_{cap,3} = \mathcal{P}_{cap,4} = \begin{Bmatrix} ae, & af, & ag, & ah, & bf, & bg, & bh, & cg, & ch \\ 1, & 1, & 1, & 1, & 2, & 1, & 1, & 1, & 1 \end{Bmatrix}$$
$$\mathcal{P}_{cap,5} = \begin{Bmatrix} e, & f, & g, & h \\ 1, & 2, & 1, & 1 \end{Bmatrix}$$

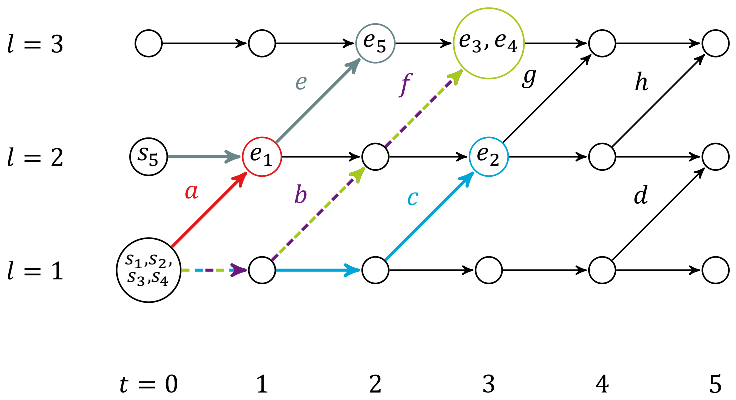


Figure 4.21: STN with two orders,$d_w = 1 \; \forall \; w \in \mathcal{W}$, with $m_b = m_f = 2$, $m_{a_i} = 1$ for $a_i \in \mathcal{A} \setminus \{b, f\}$, $m_{a_i} = \infty$ on waiting arcs. The denoted solution is SO.

We have path costs

$$\begin{aligned}
&C^a_{1,2} = 1, \quad C^b_{1,2} = 2, \quad C^c_{1,2} = 3, \quad C^d_{1,2} = 5,\\
&C^{ae}_{3,4} = 2, \quad C^{af}_{3,4} = 3, \quad C^{ag}_{3,4} = 4, \quad C^{ah}_{3,4} = 5, \quad C^{bf}_{3,4} = 3, \quad C^{bg}_{3,4} = 4, \quad C^{bh}_{3,4} = 5, \quad C^{cg}_{3,4} = 4, \quad C^{ch}_{3,4} = 5,\\
&C^e_5 = 2, \quad C^f_5 = 3, \quad C^g_5 = 4, \quad C^h_5 = 5,
\end{aligned}$$

where $C^p_{w_1,w_2}$ denotes that for both orders the same path can be traversed, because those orders have the same origin and destination location. In this example we use path flow vectors $\underline{f^p_w}$ instead of $\underline{f_p}$, because there are orders with the same origin and destination, so with the adjusted notation we can show which part of the path flow is used for which order. The path sets follow from the SO solution:

$$h_{in,1} = \begin{Bmatrix} a \\ 1 \end{Bmatrix}, \qquad h_{out,1} = \begin{Bmatrix} b, & c, & d \\ 2, & 1, & 1 \end{Bmatrix},$$
$$h_{in,2} = \begin{Bmatrix} c \\ 1 \end{Bmatrix}, \qquad h_{out,2} = \begin{Bmatrix} a, & b, & d \\ 1, & 2, & 1 \end{Bmatrix},$$
$$h_{in,3} = \begin{Bmatrix} bf \\ 1 \end{Bmatrix}, \qquad h_{out,3} = \begin{Bmatrix} ae, & af, & ag, & ah, & bf & ,bg, & bh, & cg, & ch \\ 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1 \end{Bmatrix},$$
$$h_{in,4} = \begin{Bmatrix} bf \\ 1 \end{Bmatrix}, \qquad h_{out,4} = \begin{Bmatrix} ae, & af, & ag, & ah, & bf & ,bg, & bh, & cg, & ch \\ 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1, & 1 \end{Bmatrix},$$
$$h_{in,5} = \begin{Bmatrix} e \\ 1 \end{Bmatrix}, \qquad h_{out,5} = \begin{Bmatrix} f, & g, & h \\ 2, & 1, & 1 \end{Bmatrix}.$$

We see that only orders 1 and 5 travel via their cheapest paths, so we need tolls to create a UE. Solving the NP-β gives us

$$\beta^a_1 = 1, \; \beta^c_2 = -2, \; \beta^{ae}_3 = 1, \; \beta^{ae}_4 = 1, \; \beta^e_5 = 1,$$

where for orders 3 and 4 tolls are added to paths not in the SO solution. Then the adjusted path costs become

$$C^a_{\beta 1} = 1 + 1 = 2, \quad C^a_{\beta 2} = 1, \quad C^c_{\beta 1} = 3, \quad C^c_{\beta 2} = 3 - 2 = 1, \quad C^{ae}_{\beta 3,4} = 2 + 1 = 3, \quad C^e_{\beta 5} = 2 + 1 = 3.$$

$\triangle$

4.5.2. Case 3: Finding properties of path tolls

Case 3 with two orders

Proposition 9 (Case 3 with two orders)**.** *Given is an SO problem with two orders with demand $d_w = 1$ for both orders and $m_a \geq 1$ on all traveling arcs. Assume the paths are ordered such that*

$$C_1^{r_1} \leq C_1^{r_2} \leq \cdots \leq C_1^{r_{n-1}} \leq C_1^{r_n} \text{ and } C_2^{q_1} \leq C_2^{q_2} \leq \cdots \leq C_2^{q_{m-1}} \leq C_2^{q_m},$$

where $r_j \in \mathcal{P}_1 \; \forall \, j \in \{1, \dots, n\}$ and $p_j \in \mathcal{P}_2 \; \forall \, j \in \{1, \dots, m\}$. If the SO solution contains $\underline{f_{r_i}} = 1$ with $C_1^{r_i} = C_1^{r_1}$ and $\underline{f_{q_i}} = 1$ with $C_2^{q_i} = C_2^{q_1}$, then there are no tolls needed because the SO solution is a UE.

Proof. For both orders a cheapest path can be traversed, so for every individual order the path choice is optimal, so a UE is reached. $\square$

Case 3 with multiple orders

The following proposition gives us the condition which has to hold when we only want tolls on the chosen paths, given an SO solution.

Proposition 10 (Case 3 with multiple orders)**.** *Given is an SO solution with orders $w \in \mathcal{W}$, define path sets $h_{in,w} = \left\{ \frac{p}{\underline{f_p}} \mid \underline{f_p} > 0, \; p \in \mathcal{P}_w \right\}$ and $h_{out,w} = \left\{ \frac{p}{m_p - \underline{f_p}} \mid m_p > \underline{f_p}, \; p \in \mathcal{P}_w \right\} \forall \, w \in \mathcal{W}$.*
Define $\mathcal{P}_w = \{p_1, \dots, p_n\}$ with $C_w^{p_1} \leq \dots \leq C_w^{p_n} \; \forall \, w \in \mathcal{W}$. Define for each path $p_j \in h_{in,w} \; \forall \, w \in \mathcal{W}$: $g_w^{p_j} := \{p_i \mid i < j, \; p_i \in h_{out,w}\}$, $l_w^{p_j} := \{p_i \mid i > j, \; p_i \in h_{out,w}\}$. Then to obtain a toll solution where there are only tolls on chosen paths, so $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$, the following inequality needs to hold:

$$\sum_{w \in \mathcal{W}} \sum_{q \in h_{in,w}} C_w^q \underline{f_q} \leq \sum_{w \in \mathcal{W}} \left(\sum_{q \in h_{in,w}} \min_{r \in h_{out,w}} C_w^r \underline{f_q} \right).$$

Proof. Assume $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$. As in the proof of Proposition 8, we can rewrite the toll constraints

$$\begin{aligned} \beta_w^q &\geq -C_w^q \; \forall \, q \in h_{in,w} \; \forall \, w \in \mathcal{W} \\ \beta_w^q - \beta_w^r &\leq C_w^r - C_w^q \; \forall \, q \in h_{in,w}, \; r \in h_{out,w} \; \forall \, w \in \mathcal{W} \end{aligned}$$

to

$$\begin{aligned} \beta_w^q &\leq \min_{r \in g_w^q} \{C_w^r - C_w^q\} < 0 \; \forall \, q \in h_{in,w} \; \forall \, w \in \mathcal{W} \\ \beta_w^q &\leq \min_{r \in l_w^q} \{C_w^r - C_w^q\} \; \forall \, q \in h_{in,w} \; \forall \, g_w = \emptyset, \; w \in \mathcal{W}. \end{aligned} \tag{4.22}$$

If tolls are necessary, we need both positive and negative tolls, because $\sum_{w \in \mathcal{W}} \sum_{q \in h_{in,w}} \beta_w^q \underline{f_q} = 0$. We know for paths with $g_w^q \neq \emptyset$ that $\beta_w^q < 0$, because if we do not require this, there will be unused paths with cheaper path cost than some used paths. In order to obtain this zero sum and the fact that we want tolls on chosen paths only, the following needs to hold:

$$\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\beta_w^q\underline{f_q}=\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}-\beta_w^q\underline{f_q}$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\left|\beta_w^q\underline{f_q}\right|=\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\left|\beta_w^q\underline{f_q}\right|.$$

This equality only holds if the inequalities (4.22) are satisfied when we sum over these inequalities:

$$\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\min_{r\in g_w^q}\left(C_w^q-C_w^r\right)\underline{f_q}\leq-\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\beta_w^q\underline{f_q}=\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\beta_w^q\underline{f_q}\leq\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\min_{r\in l_w^q}\left(C_w^r-C_w^q\right)\underline{f_q}$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\min_{r\in g_w^q}\left(C_w^q-C_w^r\right)\underline{f_q}\leq\sum_{w\in\mathcal{W}}\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\min_{r\in l_w^q}\left(C_w^r-C_w^q\right)\underline{f_q}.$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{q\in h_{in,w}}C_w^q\underline{f_q}\leq\sum_{w\in\mathcal{W}}\left(\sum_{\substack{q\in h_{in,w}\\ g_w^q\neq\emptyset}}\min_{r\in g_w^q}C_w^r\underline{f_q}+\sum_{\substack{q\in h_{in,w}\\ g_w^q=\emptyset}}\min_{r\in l_w^q}C_w^r\underline{f_q}\right)$$

$$\Leftrightarrow$$

$$\sum_{w\in\mathcal{W}}\sum_{q\in h_{in,w}}C_w^q\underline{f_q}\leq\sum_{w\in\mathcal{W}}\left(\sum_{q\in h_{in,w}}\min_{r\in h_{out,w}}C_w^r\underline{f_q}\right)$$

□

We can also show that Proposition 9 follows from Proposition 10. Suppose we have a problem with two orders, then due to the fact we are working with Case 3, we have $|h_{in,w}|=1$ for $w\in\{1,2\}$. Then according to Proposition 10 the inequality

$$C_1^p+C_2^q\leq\min_{r\in h_{out,1}}C_1^r+\min_{r\in h_{out,2}}C_2^r$$

has to hold.

Summarizing this section, we explored the toll solutions for Case 3 based on the initial path costs, and in Proposition 10 we see what conditions are necessary to add tolls on paths used in the SO solution only, which is equivalent to optimally minimize the NP-β.

4.6. Case 4

Case 4 is the situation in which $d_w\geq 1\ \forall\ w\in\mathcal{W}$, and $m_a\geq 1\ \forall$ arcs between different locations. or all horizontal arcs within one location, the so-called waiting arcs, we have $m_a=\infty$.
We show an example of solving a Case 4 problem in Subsection 4.6.1 and in Subsection 4.6.2 we will derive properties for path tolls.

Recall the adjusted notation of path sets for Case 3 and 4:

$$\mathcal{P}_{cap,w} = \left\{\begin{matrix} p \\ m_p \end{matrix} \mid p \in \mathcal{P}_w \right\},$$

$$h_{in,w} = \left\{\begin{matrix} p \\ \underline{f_p} \end{matrix} \mid \underline{f_p} > 0,\ p \in \mathcal{P}_w \right\},$$

$$h_{out,w} = \left\{\begin{matrix} p \\ m_p - \underline{f_p} \end{matrix} \mid m_p > \underline{f_p},\ p \in \mathcal{P}_w \right\}.$$

4.6.1. Case 4: Example

Example 17. In this example we have three orders, all with different demand: $d_1 = 3$ from location 1 to 2, $d_2 = 3$ from location 1 to 3 and $d_3 = 1$ from location 2 to 3. An SO solution is given in Figure 4.22, with s_w and e_w denoting the start end point of order w, respectively. All traveling arcs have capacity 1, except for arcs a, c and f, which have capacity $m_a = m_c = m_f = 2$, which we graphically show by multiple arcs between a pair of nodes.

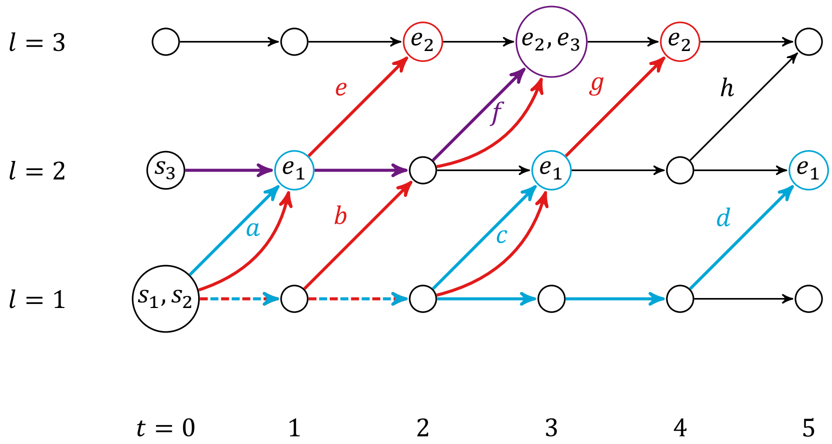


Figure 4.22: STN with two orders,$d_w \geq 1\ \forall\ w \in \mathcal{W}$, with $m_a = m_c = m_f = 2$, $m_{a_i} = 1$ for $a_i \in \mathcal{A} \setminus \{a, c, f\}$, $m_{a_i} = \infty$ on waiting arcs. The denoted solution is SO.

We have path costs

$$C_1^a = 1,\ \ C_1^b = 2,\ \ C_1^c = 3,\ \ C_1^d = 5,$$

$$C_2^{ae} = 2,\ C_2^{af} = 3,\ C_2^{ag} = 4,\ C_2^{ah} = 5,\ C_2^{bf} = 3,\ C_2^{bg} = 4,\ C_2^{bh} = 5,\ C_2^{cg} = 4,\ C_2^{ch} = 5,$$

$$C_3^e = 2,\ \ C_3^f = 3,\ \ C_3^g = 4,\ \ C_3^h = 5,$$

The path sets following from the SO solution are:

$$h_{in,1} = \left\{\begin{matrix} a, & c, & d \\ 1, & 1, & 1 \end{matrix}\right\}, \qquad h_{out,1} = \left\{\begin{matrix} a, & b, & c \\ 1, & 1, & 1 \end{matrix}\right\},$$

$$h_{in,2} = \left\{\begin{matrix} ae, & bf, & cg \\ 1, & 1, & 1 \end{matrix}\right\}, \qquad h_{out,2} = \left\{\begin{matrix} af, & ag, & ah, & bg, & bg, & ch \\ 2, & 1, & 1, & 1, & 1, & 1 \end{matrix}\right\},$$

$$h_{in,3} = \left\{\begin{matrix} f \\ 1 \end{matrix}\right\}, \qquad h_{out,3} = \left\{\begin{matrix} e, & f, & g, & h \\ 1, & 1, & 1, & 1 \end{matrix}\right\}.$$

We see that none of the orders can travel via their cheapest paths, so we need tolls to create a UE. Solving the NP-β gives us

$$\beta_1^a = 2,\ \ \beta_1^d = -1,\ \beta_2^{cg} = -1,\ \beta_2^{ae} = 1,\ \beta_3^f = -1,$$
$$\beta_1^b = 2.$$

Please not that path $b \in h_{out,1}$, so the toll on that path is not actually payed. △

4.6.2. Case 4: Finding properties of path tolls

Proposition 11 (Case 4 with multiple orders)**.** *Given is an SO solution with orders $w \in \mathcal{W}$, define path sets $h_{in,w} = \left\{ \begin{matrix} p \\ \underline{f_p} \end{matrix} \mid \underline{f_p} > 0,\ p \in \mathcal{P}_w \right\}$ and $h_{out,w} = \left\{ \begin{matrix} p \\ m_p - \underline{f_p} \end{matrix} \mid m_p > \underline{f_p},\ p \in \mathcal{P}_w \right\} \forall\ w \in \mathcal{W}$. Define $\mathcal{P}_w = \{p_1, \ldots, p_n\}$ with $C_w^{p_1} \leq \ldots \leq C_w^{p_n}\ \forall\ w \in \mathcal{W}$. Define for each path $p_j \in h_{in,w}\ \forall\ w \in \mathcal{W}$: $g_w^{p_j} := \{p_i \mid i < j,\ p_i \in h_{out,w}\}$, $l_w^{p_j} := \{p_i \mid i > j,\ p_i \in h_{out,w}\}$. Then to obtain a toll solution where there are only tolls on chosen paths, so $\sum_{w \in \mathcal{W}} \sum_{p \in h_{out,w}} |\beta_w^p| = 0$, the following inequality needs to hold:*

$$\sum_{w \in \mathcal{W}} \sum_{q \in h_{in,w}} C_w^q \underline{f_q} \leq \sum_{w \in \mathcal{W}} \left(\sum_{q \in h_{in,w}} \min_{r \in h_{out,w}} C_w^r \underline{f_q} \right).$$

Proof. Note Proposition 11 is equal to Proposition 10, except for the fact that $d_w \geq 1\ \forall\ w \in \mathcal{W}$ instead of $d_w = 1\ \forall\ w \in \mathcal{W}$, so we refer to the proof of Proposition 10. □

Summarizing this section, we explored the toll solutions for Case 4 based on the initial path costs, and in Proposition 11 we see what conditions are necessary to add tolls on paths used in the SO solution only, which is equivalent to optimally minimize the NP-β.

We give a short conclusion of this Chapter. Recall, a UE is reached when each customers can travel via their cheapest paths. When applying path based tolls, we assumed customers do not know the path costs of the initial network (and thus also do not know their cheapest paths). The tolls are used to adjust the path costs, such that we can offer the customers a choice of (tolled) paths. Then each customer chooses its cheapest tolled paths (UE), and with those choices the SO objective value is maintained.
We aimed for a toll solution in which as little as possible toll is added to unused paths in the SO solution, such that the number of paths with adjusted costs is low. However, adding tolls to paths can be unfair towards customers, because we just adjust the path cost to give customers the idea they are traveling via their cheapest paths. In the next chapter we will investigate tolls on orders, in which we remove the assumption that customers do not know the initial path costs, and therefore we are forced to find a more fair method of adding tolls.

5

Tolls on orders

In this chapter, our goal is to find a Order tolled User Equilibrium. We will use the same structure to obtain orders tolls, as we used to obtain path tolls.
Recall, we first solve the SO problem in an STN. Then in order to obtain a User Equilibrium, we adjust the costs by adding tolls. Contrary to the toll principle in the previous chapter, we do not assume that the customer is not familiar with the initial cost network. In this case, we assign tolls to the total costs of orders, and thus the value of the tolls do not influence the path choices of customers. The tolls assigned to orders will make a fair redistribution of the costs of all orders using the network.

We start by solving the SO, and then make a payment regulation such that no customer is harmed by that SO solution and thus a UE is reached. The sum of all tolls payed and received by all customers has to be zero. The way of finding tolls that give us a UE solution in a initial SO problem is described in Algorithm 3, which has the same structure as Algorithm 2 for path tolls in Chapter 4.

Recall we use the ratio $r_w = \frac{C_w}{k_w}$ to calculate how much worse the solution is for order w, compared to the cheapest costs he could have payed when he was the only order on the network. We introduce the tolled ratio $r_{\beta w} = \frac{C_{\beta w}}{k_w} = \frac{C_w + \beta_w}{k_w}$, which denotes the ratio of a tolled solution.

Algorithm 3 Calculating order tolls

1: Create SO problem:

$$
\begin{aligned}
\min \quad & \sum_{p \in \mathcal{P}} C_w^p f_p \\
\text{s.t.} \quad & x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \ \forall\, a \in \mathcal{A} \\
& \sum_{p \in \mathcal{P}_w} f_p = d_w \ \forall\, w \in \mathcal{W} \\
& x_a \leq m_a \ \forall\, a \in \mathcal{A} \\
& f_p \in \mathbb{N}_0 \ \forall\, p \in \mathcal{P} \\
& x_a \in \mathbb{N}_0 \ \forall\, a \in \mathcal{A}
\end{aligned}
$$

2: Solve SO problem, output: $C_w = \sum_{p \in \mathcal{P}_w} C_w^p \underline{f_p} \ \forall\, w \in \mathcal{W}$.

3: Create the nonlinear programming problem NP-β (where we minimize over the absolute value of the difference of ratios for all pairs of orders):

$$\min \sum_{w_1, w_2 \in \mathcal{W}} \left| r_{\beta w_1} - r_{\beta w_2} \right| = \left| \frac{C_{w_1} + \beta_{w_1}}{k_{w_1}} - \frac{C_{w_2} + \beta_{w_2}}{k_{w_2}} \right| \tag{5.1}$$

$$\text{s.t.} \sum_{w \in \mathcal{W}} \beta_w = 0 \tag{5.2}$$

$$\beta_w \geq -C_w \ \forall \ w \in \mathcal{W} \tag{5.3}$$

where the objective function minimizes the ratio differences for all pairs of orders. Note that we sum twice over the set of orders: when $w_1 = w_2$, that term of the objective becomes zero. Constraint 5.2 ensures the total toll sum to be zero, and Constraint 5.3 ensures that no path can have a negative $C_{\beta w}$ cost.

To solve the objective function of the NP-β in Algorithm 3 Step 3 more easily, we use an equivalent linear formulation to solve:

$$\min \sum_{w_1, w_2 \in \mathcal{W}} \gamma_{w_1, w_2}$$

$$\text{s.t.} \sum_{w \in \mathcal{W}} \beta_w = 0$$

$$\beta_w \geq -C_w \ \forall \ w \in \mathcal{W}$$

$$\frac{C_{w_1} + \beta_{w_1}}{k_{w_1}} - \frac{C_{w_2} + \beta_{w_2}}{k_{w_2}} \leq \gamma_{w_1, w_2} \ \forall \ w_1, w_2 \in \mathcal{W} \tag{5.4}$$

$$-\frac{C_{w_1} + \beta_{w_1}}{k_{w_1}} + \frac{C_{w_2} + \beta_{w_2}}{k_{w_2}} \leq \gamma_{w_1, w_2} \ \forall \ w_1, w_2 \in \mathcal{W} \tag{5.5}$$

$$\gamma_{w_1, w_2} \geq 0 \ \forall \ w_1, w_2 \in \mathcal{W} \tag{5.6}$$

where γ_{w_1, w_2} replaces the absolute value term $\left| \frac{C_{w_1} + \beta_{w_1}}{k_{w_1}} - \frac{C_{w_2} + \beta_{w_2}}{k_{w_2}} \right|$ in the objective function with the help of Constraints (5.4) - (5.6).

4: Solve NP-β, output: β_w.

5: Add tolls β_w to the SO problem, SO-β:

$$\min \sum_{w \in \mathcal{W}} \left(\sum_{p \in \mathcal{P}_w} C_w^p f_p + \beta_w \right)$$

$$\text{s.t.} \ x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \ \forall \ a \in \mathcal{A}$$

$$\sum_{p \in \mathcal{P}_w} f_p = d_w \ \forall \ w \in \mathcal{W}$$

$$x_a \leq m_a \ \forall \ a \in \mathcal{A}$$

$$f_p \in \mathbb{N}_0 \ \forall \ p \in \mathcal{P}$$

$$x_a \in \mathbb{N}_0 \ \forall \ a \in \mathcal{A}$$

6: Solve SO-β problem, output path flow vector $\underline{\boldsymbol{f}}$.

The desired outcome of Algorithm 3, is that the solution to the SO-β problem is equal to the initial SO problem. We will illustrate how the algorithm works in Section 5.1, with problems of Case 1. In Section 5.2 we will describe an Algorithm that splits the set of orders such that we can create a better objective function for the NP-β, to obtain a better division of toll costs. Then in Sections 5.3, 5.4 and 5.5 we will investigate the other toll on order problems, regarding Case 2,3 and 4, respectively.

5.1. Case 1

Case 1 is the situation in which $d_w = 1 \;\forall\; w \in \mathcal{W}$, and $m_a = 1 \;\forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.

In Example 18 we will show how we can verify the correctness of the tolls found. In Example 19 and Example 20 we take a closer look at the ratios en how those affect the outcome of the NP-β. In Example 21 and 22 we wll discuss the fairness of certain solutions.

Example 18. We use the STN as in Figure 4.4, again showed in Figure 5.1. The SO solution is given in Figure 5.1, with s_w denoting the start point and e_w denoting the end point for order w.

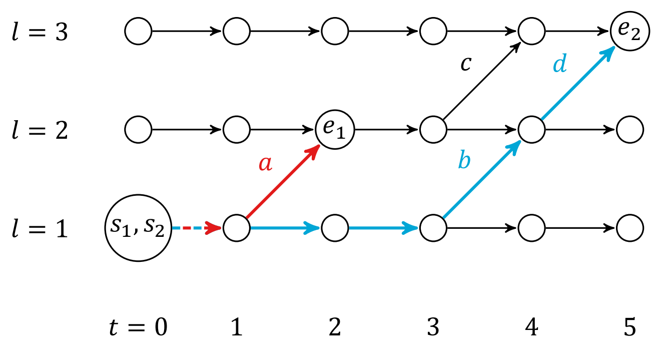


Figure 5.1: STN with two orders, $d_w = 1 \;\forall\; w \in \mathcal{W}$, with $m_a = 1$ for all arcs between two different locations, $m_a = \infty$ otherwise. The denoted solution is SO.

We have path costs $C_1^a = 2$, $C_1^b = 4$, $C_2^{ac} = 4$, $C_2^{ad} = 5$ and $C_2^{bd} = 5$. Given that for order 1 path a is used, and for order 2 path bd is used, the order costs are: $C_1 = C_1^a = 2$, $C_2 = C_2^{bd} = 5$. We see that due to capacity constraints, order 2 is not able to choose his best route in the SO and has to take path bd with cost 5.
The tolls found by the Algorithm 3 Step 3 are

$$\beta_1 = \frac{1}{3}, \quad \beta_2 = -\frac{1}{3},$$

with the objective value of the NP-β being equal to 0, which is the minimal value for an absolute value function. We show how we can derive the values of the tolls by hand, by evaluating the objective function of the NP-β and use the fact that $\beta_1 + \beta_2 = 0$:

$$\min_{\beta_1, \beta_2,\, \beta_1 = -\beta_2} \left| \frac{2+\beta_1}{2} - \frac{5+\beta_2}{4} \right| + \left| \frac{5+\beta_2}{4} - \frac{2+\beta_1}{2} \right| = \min_{\beta_1} 2 \cdot \left| 1 + \frac{\beta_1}{2} - \frac{5}{4} - \frac{-\beta_1}{4} \right| = \min_{\beta_1} 2 \cdot \left| -\frac{1}{4} + \frac{3}{4}\beta_1 \right|.$$

The minimum value of an absolute value is zero, so we can check if this can be the case:

$$0 = \left| -\frac{1}{4} + \frac{3}{4}\beta_1 \right| \Leftrightarrow \frac{1}{4} = \frac{3}{4}\beta_1 \Leftrightarrow \frac{1}{3} = \beta_1 \Rightarrow \beta_2 = -\frac{1}{3}.$$

Then the tolled costs become $C_{\beta 1} = 2\frac{1}{3}$, $C_{\beta 2} = 4\frac{2}{3}$, so both ratios are $r_{\beta w} = 1\frac{1}{6}$. $\triangle$

Example 19. We use the STN in Figure 5.2.

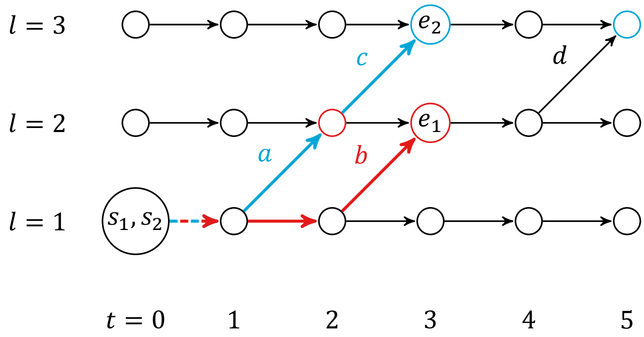


Figure 5.2: STN with two orders, $d_w = 1\ \forall\ w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

We will now observe which toll we need to add to obtain a UE that equals the initial SO. The SO solution is given in Figure 5.2, with s_w and e_w denoting the start and end point of order w, respectively.
We have path costs $C_1^a = 2,\ C_1^b = 3,\ C_2^{ac} = 3,\ C_2^{ad} = 5$ and $C_2^{bd} = 5$.
The solution given by Algorithm 3 Step 4 is

$$\beta_1 = -\frac{3}{5},\ \beta_2 = \frac{3}{5},$$

and thus the order costs are $C_{\beta 1} = 2\frac{2}{5},\ C_{\beta 2} = 3\frac{3}{5}$.
We can see in Table 5.1 that indeed the ratio of current cost compared to the cheapest path cost is

Table 5.1: Example of Figure 5.2: Ratios $r_{\beta w} = (C_w + \beta_w)/k_w$ per toll

	Order 1	$C_{\beta 1}$	$r_{\beta 1}$	Order 2	$C_{\beta 2}$	$r_{\beta 2}$
k_w	2			3		
C_w (SO)	3		1.5	3		1
β_w	−0.1	2.9	1.45	0.1	3.1	1.0333
	−0.2	2.8	1.4	0.2	3.2	1.0667
	−0.3	2.7	1.35	0.3	3.3	1.1
	−0.4	2.6	1.3	0.4	3.4	1.1333
	−0.5	2.5	1.25	0.5	3.5	1.1667
NP-β toll:	−0.6	2.4	1.2	0.6	3.6	1.2
	−0.7	2.3	1.15	0.7	3.7	1.2333
	−0.8	2.2	1.1	0.8	3.8	1.2667
	−0.9	2.1	1.05	0.9	3.9	1.3
	−1	2	1	1	4	1.3333

equal for both orders, namely $r_1 = r_2 = 1\frac{1}{5}$. △

Example 20. In this example there are three orders $w = \{1, 2, 3\}$, with $d_w = 1\ \forall\ w \in \mathcal{W}$. The STN and the SO solution are given in Figure 5.3 with s_w and e_w denoting the start and end point of order w, respectively.

The order costs of the SO are $C_1 = C_1^a = 2,\ C_2 = C_2^{bd} = 5,\ C_3 = C_3^c = 3$.
We solve the NP-β manually to see how the solution is found:

$$\min_{\substack{\boldsymbol{\beta},\\ \beta_1+\beta_2+\beta_3=0}} \left|\frac{2+\beta_1}{2} - \frac{5+\beta_2}{3}\right| + \left|\frac{2+\beta_1}{2} - \frac{3+\beta_3}{3}\right| + \left|\frac{5+\beta_2}{3} - \frac{2+\beta_1}{2}\right| + \left|\frac{5+\beta_2}{3} - \frac{3+\beta_3}{3}\right| +$$
$$\left|\frac{3+\beta_3}{3} - \frac{2+\beta_1}{2}\right| + \left|\frac{3+\beta_3}{3} - \frac{5+\beta_2}{3}\right| =$$

$$= \min_{\substack{\boldsymbol{\beta},\\ \beta_1+\beta_2+\beta_3=0}} 2\left(\left|\frac{2+\beta_1}{2} - \frac{5+\beta_2}{3}\right| + \left|\frac{2+\beta_1}{2} - \frac{3+\beta_3}{3}\right| + \left|\frac{5+\beta_2}{3} - \frac{3+\beta_3}{3}\right|\right)$$

$$= \min_{\substack{\boldsymbol{\beta},\\ \beta_1+\beta_2+\beta_3=0}} 2\left(\left|\frac{-4+3\beta_1-2\beta_2}{6}\right| + \left|\frac{3\beta_1-2\beta_3}{6}\right| + \left|\frac{2+\beta_2-\beta_3}{3}\right|\right)$$

This objective has value zero if all individual absolute values are equal to zero and we also want the total toll sum zero constraint to hold, so we obtain four equalities:

$$\begin{aligned} 3\beta_1 - 2\beta_2 &= 4 \\ 3\beta_1 - 2\beta_3 &= 0 \\ \beta_2 - \beta_3 &= -2 \\ \beta_1 + \beta_2 + \beta_3 &= 0 \end{aligned} \quad \Leftrightarrow \quad \begin{bmatrix} 3 & -2 & 0 \\ 3 & 0 & -2 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} 4 \\ 0 \\ -2 \\ 0 \end{bmatrix},$$

which gives us the toll solution as in Table 5.2: $\beta_1 = \frac{1}{2}$, $\beta_2 = -1\frac{1}{4}$, $\beta_3 = \frac{3}{4}$, and order costs $C_{\beta 1} = 2\frac{1}{2}$, $C_{\beta 2} = 3\frac{3}{4}$, $C_{\beta 3} = 3\frac{3}{4}$, so all orders have ratio $r_{\beta w} = 1\frac{1}{4}$.

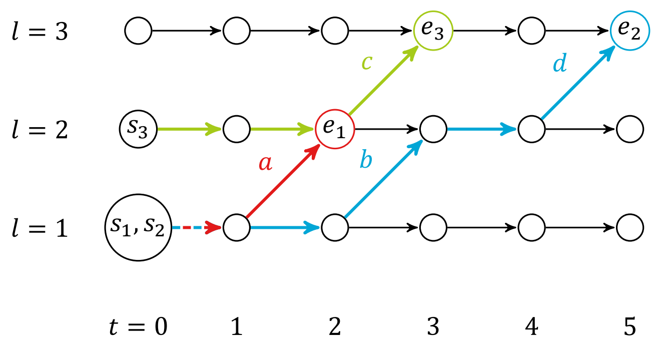


Figure 5.3: STN with three orders, $d_w = 1 \; \forall \; w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

Table 5.2: Ratios $r_{\beta w} = (C_w + \beta_w)/k_w$ per toll for Figure 5.3

	Order 1	$C_{\beta 1}$	$r_{\beta 1}$	Order 2	$C_{\beta 2}$	$r_{\beta 2}$	Order 3	$C_{\beta 3}$	$r_{\beta 3}$
k_w	2			3			3		
C_w (SO)	2		1	5		$1\frac{2}{3}$	3		1
NP-β toll: β_w	$\frac{1}{2}$	$2\frac{1}{2}$	$1\frac{1}{4}$	$-1\frac{1}{4}$	$3\frac{3}{4}$	$1\frac{1}{4}$	$\frac{3}{4}$	$3\frac{3}{4}$	$1\frac{1}{4}$

In Table 5.2 we can compare the ratios of the SO solution to those of the tolled solution. We see that the extra costs in the SO solution of order 2, are divided over all orders in the tolled solution. △

Example 21. Here we have three orders, with one order not interfering with the other two orders regarding their cheapest paths. The STN is given in Figure 5.4.

The SO solution is given in Figure 5.4, the order tolls are

$$\beta_1 = -\frac{2}{3}, \; \beta_2 = \frac{5}{12}, \; \beta_3 = \frac{1}{4}.$$

The question now is, is this solution fair? Order 3 does not cause any bottlenecks for the other orders, so why would this order have to pay for the problems of other orders? If we claim that order 3 does not have to pay/receive any toll, so $\beta_3 = 0$, and solve the NP-β again, we obtain the "fair" toll solution as in

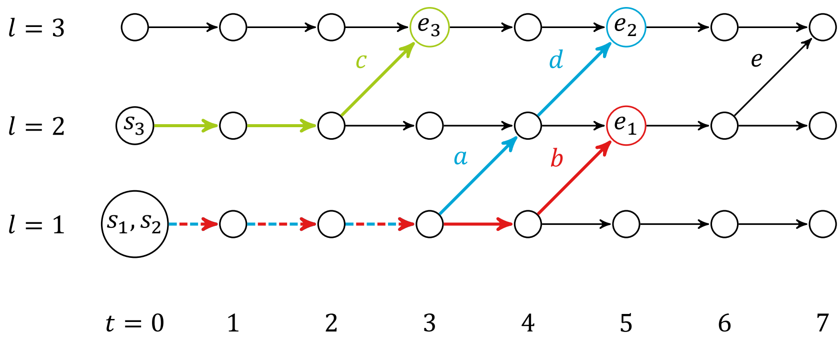


Figure 5.4: STN with three orders, $d_w = 1 \; \forall \; w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

Table 5.3: Ratios $r_{\beta w} = (C_w + \beta_w)/k_w$ per toll for Figure 5.4

	Order 1	$C_{\beta 1}$	$r_{\beta 1}$	Order 2	$C_{\beta 2}$	$r_{\beta 2}$	Order 3	$C_{\beta 3}$	$r_{\beta 3}$
k_w	4			5			3		
C_w (SO)	5		$1\frac{1}{4}$	5		1	3		1
NP-β toll: β_w	$-\frac{2}{3}$	$4\frac{1}{3}$	$1\frac{1}{12}$	$\frac{5}{12}$	$5\frac{5}{12}$	$1\frac{1}{12}$	$\frac{1}{4}$	$3\frac{1}{4}$	$1\frac{1}{12}$
"Fair" toll: β_w	$-\frac{5}{9}$	$4\frac{4}{9}$	$1\frac{1}{9}$	$\frac{5}{9}$	$5\frac{5}{9}$	$1\frac{1}{9}$	0	3	1

Table 5.3, in which order 1 receives $\frac{5}{9}$ from order 2. Then both order 1 and 2 have a ratio $r_1 = r_2 = 1\frac{1}{9}$, and $r_3 = 1$. We will discuss this fairness in the next example and in Section 5.2. △

Example 22. This is another example with three orders. The STN is given in Figure 5.5.

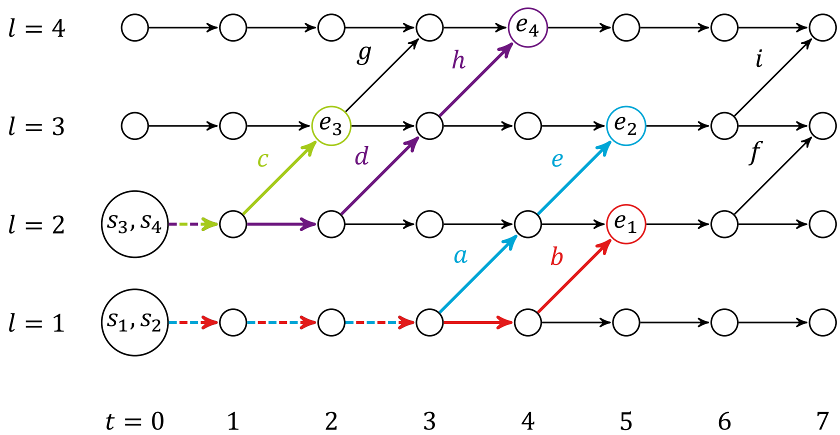


Figure 5.5: STN with four orders, $d_w = 1 \; \forall \; w \in \mathcal{W}$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

The SO solution is given in Figure 5.5, the order tolls following from Algorithm 3 are

$$\beta_1 = -\frac{3}{7}, \; \beta_2 = \frac{5}{7}, \; \beta_3 = \frac{2}{7}, \; \beta_4 = -\frac{4}{7},$$

so the order costs are $C_{\beta 1} = 4\frac{4}{7}$, $C_{\beta 2} = 5\frac{5}{7}$, $C_{\beta 3} = 2\frac{2}{7}$, $C_{\beta 4} = 3\frac{3}{7}$.
The question now is, is this solution fair? It looks like optional paths for order 1 and 2 are not interfering with optional paths for order 3 and 4. Therefore we calculate "fair" tolls, and for that we need to divide all orders in different sets, each set with their own joint bottleneck arcs.

We first compare the path costs per order:

$$
\begin{array}{llllllllllll}
\mathcal{P}_1 = \{a,\ b\}, & C_1^a & < & C_1^b, & & & & & & & & \\
\mathcal{P}_2 = \{ae,\ af,\ bf\}, & C_2^{ae} & < & C_2^{af} & = & C_2^{bf}, & & & & & & \\
\mathcal{P}_3 = \{c,\ d,\ e,\ f\}, & C_3^{c} & < & C_3^{d} & < & C_3^{e} & < & C_3^{f}, & & & & \\
\mathcal{P}_4 = \{cg,\ ch,\ ci,\ dh,\ di,\ ei\}, & C_4^{cg} & < & C_4^{ch} & = & C_4^{dh} & < & C_4^{ci} & = & C_4^{di} & = & C_4^{ei}.
\end{array}
$$

We can easily see in Figure 5.5 that for order 1 and 2, arc a is in both cheapest paths (those are path a and path ae) and for order 3 and 4 this is the case for arc c (paths c and cg). Now instead of recalculating the order costs by dividing the total cost per ratio over all orders, we can first divide the orders into two subgroups in this case, where orders in one subgroup have shared arcs in optional paths. So we create one subset for arc a with orders 1 and 2, and one for arc c for orders 3 and 4.

Table 5.4: Ratios $r_{\beta w} = (C_w + \beta_w)/k_w$ per toll for Figure 5.5

	Order 1	$C_{\beta 1}$	$r_{\beta 1}$	Order 2	$C_{\beta 2}$	$r_{\beta 2}$	Order 3	$C_{\beta 3}$	$r_{\beta 3}$	Order 4	$C_{\beta 4}$	$r_{\beta 4}$
k_w	4			5			2			3		
C_w	5		$1\frac{1}{4}$	5		1	2		1	4		$1\frac{1}{3}$
NP-β toll: β_w	$-\frac{3}{7}$	$4\frac{4}{7}$	$1\frac{1}{7}$	$\frac{5}{7}$	$5\frac{5}{7}$	$1\frac{1}{7}$	$\frac{2}{7}$	$2\frac{2}{7}$	$1\frac{1}{7}$	$-\frac{4}{7}$	$3\frac{3}{7}$	$1\frac{1}{7}$
"Fair" toll: β_w	$-\frac{5}{9}$	$4\frac{4}{9}$	$1\frac{1}{9}$	$\frac{5}{9}$	$5\frac{5}{9}$	$1\frac{1}{9}$	$\frac{2}{5}$	$2\frac{2}{5}$	$1\frac{1}{5}$	-$\frac{2}{5}$	$3\frac{3}{5}$	$1\frac{1}{5}$

Then we can calculate "fair" tolls, as given in the last column in Table 5.4. We calculated these tolls by changing the objective of the NP-β to:

$$\min_{\boldsymbol{\beta}} \left| \frac{C_1 + \beta_1}{k_1} - \frac{C_2 + \beta_2}{k_2} \right| + \left| \frac{C_3 + \beta_3}{k_3} - \frac{C_4 + \beta_4}{k_4} \right|.$$

such that only the difference between ratios for orders in one subgroup are minimized. △

5.2. Finding connected components in STN

We use the following algorithm, Algorithm 4, to find all connected components in the STN to divide the order costs more fairly over the orders. We first give a short outline of how the algorithm works.

We search for the connected components by constructing a graph G consisting of orders $w \in \mathcal{W}$, where two orders in this graph can share an arc when those orders contain a joint bottleneck. To obtain these arcs, we first find the cheapest paths per order (in Step 4). For each arc a we check if this arc is a bottleneck, that is when the amount of cheapest paths on that arc is higher than the capacity of that arc (this happens in Step 7), assuming every order travels via its cheapest path.

If in this step bottlenecks are found, we create bottleneck sets in which paths (bottleneck sets η_a) and orders (bottleneck sets Γ_a) are listed per bottleneck arc a. We also add an arc in G between each pair of orders that are in the same bottleneck set. Then for each cheapest path of an order that is in the SO solution, we fix this path (so state that this path is taken by that order) and then recalculate the cheapest paths for all remaining orders (Step 19). Then if new bottlenecks arise, we add arcs between each pair of orders in this bottleneck set to the graph G. We iterate the fixing process until no new bottlenecks arise. Then in Step 51 we start finding the connected components in graph G.

Algorithm 4 Finding connected components in STN

Let $\mathcal{Q}_w = \emptyset \ \forall \ w \in \mathcal{W}$ be the set of cheapest paths per order w, $\mathcal{A}_\eta = \emptyset$ the bottleneck set.
Create graph G with nodes $V = \mathcal{W}$ and arc set $E = \emptyset$.
for $w \in \mathcal{W}$ **do**
  Find cheapest paths $q_{w,n},\ n \in \{1, \dots, d_w\}$ (for each unit of flow a cheapest path is given), $q_{w,n} \in \mathcal{P}_w$.
  Add paths $q_{w,n}$ to $\mathcal{Q}_w$.
end for
for $a \in \mathcal{A}$ **do**
  if $o_a > m_a$ (o_a the occupancy on arc a) **then**
    Create bottleneck set: $\eta_a = \left\{q_{w,n} \mid \delta_{a q_{w,n}} = 1,\ q_{w,n} \in \mathcal{Q}_w,\ \forall \ w \in \mathcal{W}\right\}$.
    Create order set: $\Gamma_a = \{w \mid q_{w,n} \in \eta_a,\ q_{w,n} \in \mathcal{Q}_w\}$.
    $\mathcal{A}_\eta := \mathcal{A}_\eta \cup a$.
    for $w_1, w_2 \in \Gamma_a$ **do**
      Add $e = (w_1, w_2)$ to E.
    end for
  else
    $\eta_a = \emptyset$.
  end if
end for
for $a_1 \in \mathcal{A}_\eta$ **do**
  for $q_{w,n} \in \eta_{a_1}$ and $q_{w,n} \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ **do**
    Fix $q_{w,n}$: update $o_a = \delta_{a q_{w,n}} f_{q_{w,n}} \ \forall \ a \in \mathcal{A}$.
    Recalculate the cheapest paths for all remaining orders.
    Update $\mathcal{Q}_w$.
    Define $l = (q_{w,n})$ (a list of all previous fixed paths) and $\mathcal{A}_\eta^l = \emptyset$.
    if $o_{a_2} > m_{a_2}$ for $a_2 \notin \mathcal{A}_\eta$ **then**
      Create new bottleneck set $\eta_{a_2} = \left\{q_{w,n} \mid \delta_{a_2 q_{w,n}} = 1,\ q_{w,n} \in \mathcal{Q}_w,\ \forall \ w \in \mathcal{W}\right\}$.
      Create order set: $\Gamma_{a_2} = \{w \mid q_{w,n} \in \eta_{a_2},\ q_{w,n} \in \mathcal{Q}_w\}$.
      Define $\mathcal{A}_\eta^l := \mathcal{A}_\eta^l \cup a_2$.
      for $w_1, w_2 \in \Gamma_{a_2}$ **do**
        Add $e = (w_1, w_2)$ to E.
      end for
    end if
    for $a_2 \in \mathcal{A}_\eta^l$ **do**
      for $r_{w,n} \in \eta_{a_2}$ and $r_{w,n} \in \bigcup_{w \in \mathcal{W}} h_{in,w}$ **do**
        Fix $r_{w,n}$: update $o_a = \sum_{q_{w,n} \in \text{fixed paths}} \delta_{a q_{w,n}} f_{q_{w,n}} \ \forall \ a \in \mathcal{A}$
        Recalculate the cheapest paths for all remaining orders.
        Update $\mathcal{Q}_w$.
        if $o_{a_2} > m_{a_2}$ for $a_2 \notin \mathcal{A}_\eta$ **then**
          Create new bottleneck set $\eta_{a_2} = \left\{q_{w,n} \mid \delta_{a_2 q_{w,n}} = 1,\ q_{w,n} \in \mathcal{Q}_w,\ \forall \ w \in \mathcal{W}\right\}$.
          Create order set: $\Gamma_{a_2} = \{w \mid q_{w,n} \in \eta_{a_2},\ q_{w,n} \in \mathcal{Q}_w\}$.
          $l := (l, q_{w,n})$.
          Define $\mathcal{A}_\eta^l := \mathcal{A}_\eta^l \cup a_2$.
          for $w_1, w_2 \in \Gamma_{a_2}$ **do**
            Add $e = (w_1, w_2)$ to E.
          end for
        end if
      end for
    end for
  end for
end for

$s = 1$.
$\mathcal{V}_s$ is the set of visited nodes in connected component s.
for $w_1 \in \mathcal{V} \setminus \bigcup_{1 \le j \le s} \mathcal{V}_j$ **do**
    $\mathcal{V}_s := \{w_1\}$.
    for $w_2 \in \bigcup_{v \in \mathcal{V}_s} N_G(v) \setminus \bigcup_{1 \le j \le s} \mathcal{V}_j$ **do**
        $\mathcal{V}_s := \mathcal{V}_s \cup w_2$.
    end for
    $s := s + 1$.
end for
$k := s - 1$ are the number of connected components of G.
$\mathcal{V}_s$ are the connected components of graph G.

With these connected component sets we can create a new NP-β for Algorithm 3 for finding the order tolls:

$$
\begin{aligned}
\min \ & \sum_{s=1}^{k} \sum_{w_1, w_2 \in \mathcal{V}_s} \left| \frac{C_{w_1} + \beta_{w_1}}{k_{w_1}} - \frac{C_{w_2} + \beta_{w_2}}{k_{w_2}} \right| \\
\text{s.t.} \ & \sum_{w \in \mathcal{W}} \beta_w = 0 \\
& \sum_{w \in \mathcal{V}_s} \beta_w = 0 \ \forall\ 1 \le s \le k \\
& \beta_w \ge -C_w \ \forall\ w \in \mathcal{W}
\end{aligned}
$$

This new objective function only compares the new costs of orders that are in the same connected component, and for each connected component the tolls sum up to zero such that orders only pay/receive for bottlenecks that influence the route choice for them.

We use the following examples to explain Algorithm 4. In Example 23 we consider a problem with three orders, in Example 24 a problem with four orders is considered.

Example 23. We consider a problem of Case 1. In Figure 5.6 an SO solution of an STN with three orders is given, $\mathcal{W} = \{1,\ 2,\ 3\}$. We can use the algorithm to show that in this case all orders are interfering with each other.

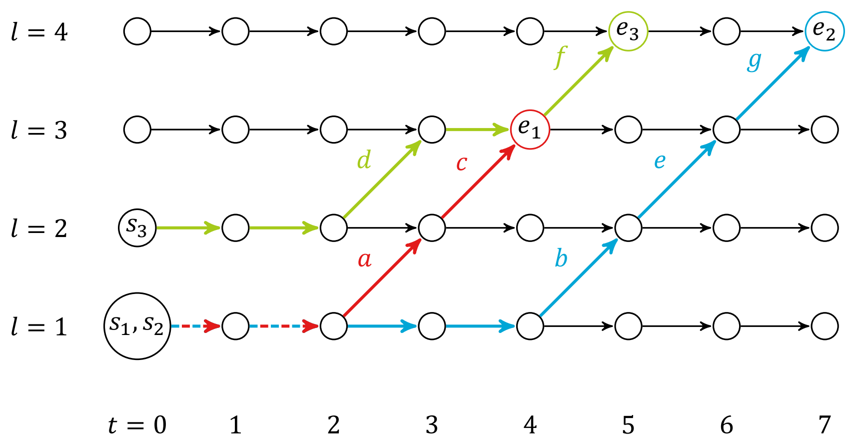


Figure 5.6: STN with three orders, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

We start with empty sets $Q_w \; \forall \; w \in \mathcal{W}$ and $\mathcal{A}_\eta$ and we create the graph $G = (V, E)$ with $V = \mathcal{W}$ and $E = \emptyset$.
Applying Algorithm 4, Step 4 gives cheapest paths $q_{1,1} = ac,\; q_{2,1} = acf,\; q_{3,1} = df$.
Step 5 gives $Q_1 = \{ac\},\; Q_2 = \{acf\},\; Q_3 = \{df\}$.
Step 9 and 10 give bottleneck arcs and order sets per bottleneck: $\eta_a = \{ac,\; acf\}$, $\Gamma_a = \{1,2\}$; $\eta_c = \{ac,\; acf\}$, $\Gamma_c = \{1,2\}$ and $\eta_f = \{acf,\; df\}$, $\Gamma_f = \{2,3\}$.
Step 11 gives bottleneck set $\mathcal{A}_\eta = \{a,\; c,\; f\}$.
We add edges $(1,2)$ and $(2,3)$ to graph G.

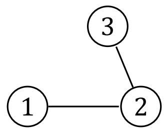


Figure 5.7: Graph G.

We see that all orders are connected: $\mathcal{V}_1 = \{1,\; 2,\; 3\}$. Then the NP-$\beta$ becomes

$$\begin{aligned}
& \min \left| \frac{C_1 + \beta_1}{k_1} - \frac{C_2 + \beta_2}{k_2} \right| + \left| \frac{C_1 + \beta_1}{k_1} - \frac{C_3 + \beta_3}{k_3} \right| + \left| \frac{C_2 + \beta_2}{k_2} - \frac{C_3 + \beta_3}{k_3} \right| \\
& \text{s.t.} \;\; \beta_1 + \beta_2 + \beta_3 = 0 \\
& \qquad \beta_w \geq -C_w \; \forall \; w \in \mathcal{W} \\
\Leftrightarrow \quad & \min \left| \frac{4 + \beta_1}{4} - \frac{7 + \beta_2}{5} \right| + \left| \frac{4 + \beta_1}{4} - \frac{5 + \beta_3}{5} \right| + \left| \frac{7 + \beta_2}{5} - \frac{5 + \beta_3}{5} \right| \\
& \text{s.t.} \;\; \beta_1 + \beta_2 + \beta_3 = 0 \\
& \qquad \beta_1 \geq -4 \\
& \qquad \beta_2 \geq -7 \\
& \qquad \beta_2 \geq -5
\end{aligned}$$

which has an optimal solution with $\beta_1 = \frac{4}{7}$, $\beta_2 = -\frac{9}{7}$ and $\beta_3 = \frac{5}{7}$ and thus tolled order costs $C_{\beta 1} = 4\frac{4}{7}$, $C_{\beta 2} = 5\frac{5}{7}$ and $C_{\beta 3} = 5\frac{5}{7}$. So all orders have ratio $r_{\beta w} = 1\frac{1}{7} \; \forall \; w \in \mathcal{W}$. $\triangle$

Example 24. We use the same STN and orders as in Example 22 (so a Case 1 problem), but with arc d removed. We see that a whole new SO solution appears, compared to the previous network. This SO solution is given in Figure 5.8.

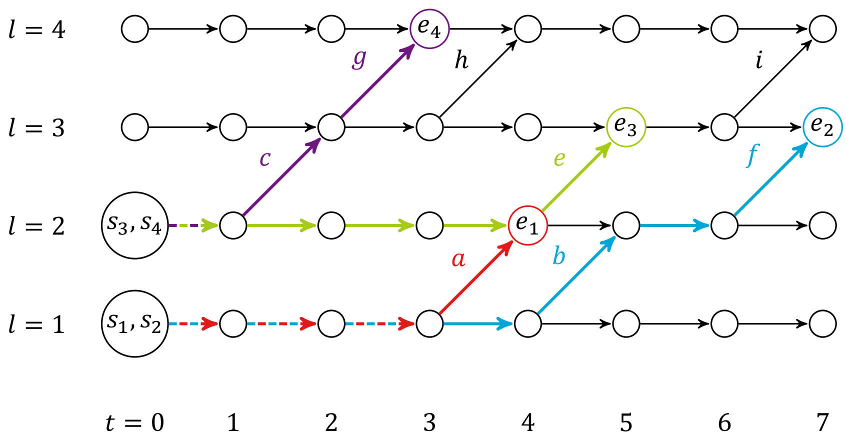


Figure 5.8: STN with four orders, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

We will now apply Algorithm 4 on this network. We start with empty sets Q_w $\forall$ $w \in \mathcal{W}$ and $\mathcal{A}_\eta$ and we create the graph $G = (V, E)$ with $V = \mathcal{W} = \{1,\ 2,\ 3,\ 4\}$ and $E = \emptyset$.
Applying Algorithm 4, Step 4 gives cheapest paths $q_{1,1} = a,\ q_{2,1} = ae,\ q_{3,1} = c,\ q_{4,1} = cg$.
Step 5 gives $Q_1 = \{a\},\ Q_2 = \{ae\},\ Q_3 = \{c\},\ Q_4 = \{cg\}$.
Step 9 and 10 give bottleneck arcs and order sets per bottleneck: $\eta_a = \{a,\ ae\}$, $\Gamma_a = \{1, 2\}$; $\eta_c = \{c,\ cg\}$, $\Gamma_c = \{3, 4\}$.
Step 10 gives $\mathcal{A}_\eta = \{a,\ c\}$.
We add edges $(1, 2)$ and $(3, 4)$ to graph G.

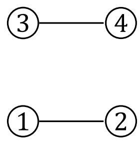


Figure 5.9: Graph G.

Then we want to join order sets in which one order can influence the path choice of another order in the same set.
We want to see if fixing one path for some order can create extra bottlenecks for other orders, which starts in the Algorithm in Step 19.
We take $a \in \mathcal{A}_\eta$ and path $a \in \eta_a$. Note that path $a \in h_{in,1}$.
We fix path a for order 1 in the STN and recalculate new cheapest paths for all other orders and we want to see if new bottlenecks arise. Now because arc a is used for path a, order 2 has to travel via another path than ae. Its new cheapest path now is $q_{2,1} = bf$. No new bottleneck arc arises, so we check the next path.
We now take $c \in \mathcal{A}_\eta$ and path $c \in \eta_c$, $c \in h_{in,3}$ and fix this path.
Then order 4 has to find a new cheapest path: $q_{4,1} = ei$. Then arc e is a bottleneck, because both $q_{2,1} = ae$ and $q_{4,1}$ wants to travel via e. So we create a new bottleneck set $\eta_e = \{ae,\ ei\}$ and order set $\Gamma_e = \{2,\ 4\}$. So we add edge $(2, 4)$ to graph G.

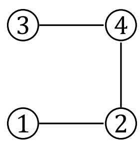


Figure 5.10: Graph G.

We now see that all orders are connected.
Then the tolls become $\beta_1 = 1\frac{3}{7}$, $\beta_2 = -\frac{3}{14}$, $\beta_3 = -1\frac{3}{7}$, $\beta_4 = 2\frac{1}{7}$ and thus the tolled order costs become $C_{\beta 1} = 5\frac{3}{7}$, $C_{\beta 2} = 6\frac{11}{14}$, $C_{\beta 3} = 3\frac{4}{7}$, and $C_{\beta 4} = 5\frac{1}{7}$, so all orders have ratio $r_{\beta w} = 1\frac{5}{7}$ $\forall$ $w \in \mathcal{W}$. $\triangle$

5.3. Case 2

Case 2 is the situation in which $d_w \geq 1$ $\forall$ $w \in \mathcal{W}$, and $m_a = 1$ $\forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
We also use Algorithm 4 to find a division of the extra costs in the system in a fair way over the orders.

Example 25. In this example two orders both with demand $d_w = 2$ are given. The SO solution is given in Figure 5.11. We see that both orders cannot travel via their cheapest paths.

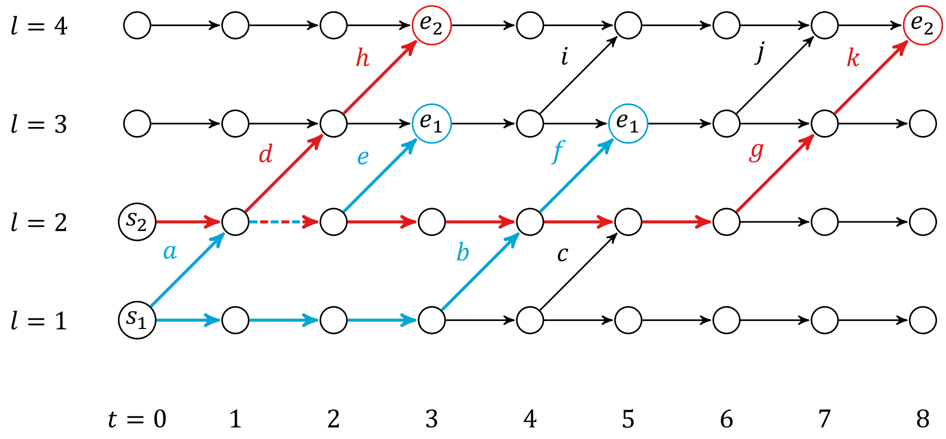


Figure 5.11: STN with two orders, $d_1 = d_2 = 2$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

The SO costs are $C_1 = 8$, $C_2 = 11$. Their cheapest paths are $k_1 = 7$ and $k_2 = 8$. Algorithm 4 gives that both orders are in one connected component, and the NP-β gives $\beta_1 = \frac{13}{15}$, $\beta_2 = -\frac{13}{15}$, which give adjusted costs $C_{\beta 1} = 8\frac{13}{15}$, $C_{\beta 2} = 10\frac{2}{15}$. Then we can verify that $r_{\beta 1} = r_{\beta 2} = \frac{19}{15}$. △

5.4. Case 3

Case 3 is the situation in which $d_w = 1\ \forall\ w \in \mathcal{W}$, and $m_a \geq 1\ \forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$. We also use Algorithm 4 to find the connected components to divide the extra costs (compared to the cheapest path costs) in the system in a fair way over the orders.

Example 26. We use the same STN as in Example 16. In this example we have 5 orders all with demand 1: Order 1,2, 3 and for start at location 1, order 5 starts at location 2. Order 1 and 2 end at location 2 and the others at location 3. The SO solution is given in Figure 4.21, with s_w and e_w denoting the start end point of order w, respectively. All traveling arcs have capacity 1, except for arcs b and f, which have capacity $m_b = m_f = 2$.

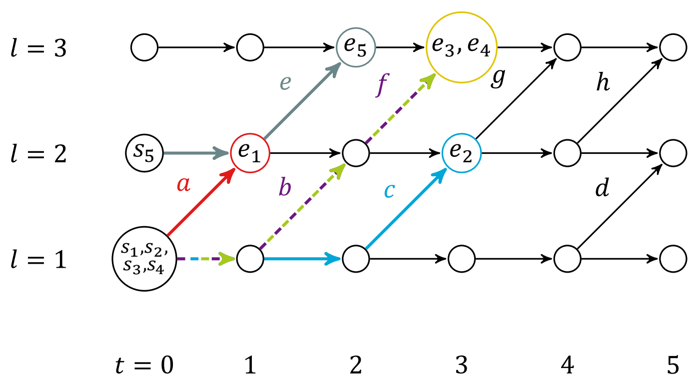


Figure 5.12: STN with two orders, $d_w = 1\ \forall\ w \in \mathcal{W}$, with $m_b = m_f = 2$, $m_{a_i} = 1$ for all other arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

We have path costs

$C^{a}_{1,2} = 1,\quad C^{b}_{1,2} = 2,\quad C^{c}_{1,2} = 3,\quad C^{d}_{1,2} = 5,$
$C^{ae}_{3,4} = 2,\quad C^{af}_{3,4} = 3,\quad C^{ag}_{3,4} = 4,\quad C^{ah}_{3,4} = 5,\quad C^{bf}_{3,4} = 3,\quad C^{bg}_{3,4} = 4,\quad C^{bh}_{3,4} = 5,\quad C^{cg}_{3,4} = 4,\quad C^{ch}_{3,4} = 5,$
$C^{e}_{5} = 2,\quad C^{f}_{5} = 3,\quad C^{g}_{5} = 4,\quad C^{h}_{5} = 5,$

So we have the following path sets per order with corresponding capacities:

$$\mathcal{P}_{cap,1} = \mathcal{P}_{cap,2} = \begin{Bmatrix} a, & b, & c, & d \\ 1, & 2, & 1, & 1 \end{Bmatrix}$$

$$\mathcal{P}_{cap,3} = \mathcal{P}_{cap,4} = \begin{Bmatrix} ae, & af, & ag, & ah, & bf, & bg, & bh, & cg, & ch \\ 1, & 1, & 1, & 1, & 2, & 1, & 1, & 1, & 1 \end{Bmatrix}$$

$$\mathcal{P}_{cap,5} = \begin{Bmatrix} e, & f, & g, & h \\ 1, & 2, & 1, & 1 \end{Bmatrix}$$

where $C^{p}_{w_1,w_2}$ denotes that for both orders the same path can be traversed, because those orders have the same origin and destination location.

Solving the SO problem gives us the order costs

$$C_1 = 1,\ C_2 = 3,\ C_3 = C_4 = 3,\ C_5 = 2,$$

while the cheapest costs are

$$k_1 = 1,\ k_2 = 1,\ k_3 = k_4 = 2,\ C_5 = 2.$$

We see that only orders 1 and 5 travel via their cheapest paths, so we need tolls to create a UE. Algorithm 4 gives us one connected components so we can find a solution for the NP-β:

$$\beta_1 = \frac{1}{2},\quad \beta_2 = -1\frac{1}{2},\quad \beta_5 = 1.$$

Then the adjusted order costs become

$$C_{\beta 1} = 1\frac{1}{2},\quad C_{\beta 2} = 1\frac{1}{2},\quad C_{\beta 3} = 3,\quad C_{\beta 4} = 3,\quad C^{e}_{\beta 5} = 1,$$

so all ratios are $r_{\beta w} = 1\frac{1}{2}\ \forall\, w \in \{1,\dots,5\}$. △

5.5. Case 4

Case 4 is the situation in which $d_w \geq 1\ \forall\, w \in \mathcal{W}$, and $m_a \geq 1\ \forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
We also use Algorithm 4 to find the connected components to divide the extra costs (compared to the cheapest path costs) in the system in a fair way over the orders.

Example 27. We use the STN as in Figure 5.13. There are four orders, with demands $d_1 = 3$, $d_2 = 2$, $d_3 = 2$ and $d_4 = 4$. All diagonal arcs have capacity constraints: $m_a = 4$, $m_b = 4$, $m_c = 2$, $m_d = 2$, $m_e = 2$, $m_f = 2$, $m_g = 3$, $m_h = 3$, $m_i = 1$, $m_j = 2$, $m_k = 2$, $m_l = 3$ and $m_m = 2$.
The SO solution is given in Figure 5.13, with s_w and e_w denoting the start and end point for order w, respectively. Note that order 3 starts at $t = 3$, location $l = 2$.

We have path capacities:

$$\mathcal{P}_{cap,1} = \begin{Bmatrix} ad, & ae, & af, & ag, & ah, & bf, & bg, & bh, & ch \\ 2, & 2, & 2, & 3, & 3, & 2, & 3, & 3, & 2 \end{Bmatrix},$$

$$\mathcal{P}_{cap,2} = \begin{Bmatrix} adi, & adj, & adk, & adl, & adm, & aek, & ael, & aem, & afj, & afk, & afl, & afm, & agl, & agm, & ahm \\ 1, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 3, & 2, & 2 \end{Bmatrix},$$

$$\mathcal{P}_{cap,3} = \begin{Bmatrix} g, & h \\ 3, & 3 \end{Bmatrix},$$

$$\mathcal{P}_{cap,4} = \begin{Bmatrix} di, & dj, & dk, & dl, & dm, & ek, & el, & em, & fj, & fk, & fl, & fm, & gl, & gm, & hm \\ 1, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 3, & 2, & 2 \end{Bmatrix}.$$

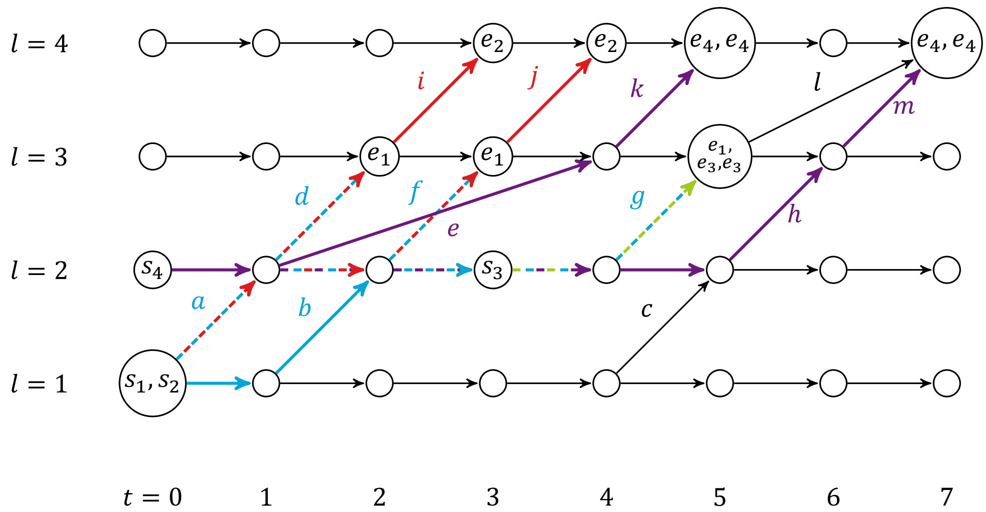


Figure 5.13: STN with four orders, $d_1 = 3$, $d_2 = d_3 = 2$, $d_4 = 4$, Capacities $m_a = 4$, $m_b = 4$, $m_c = 2$, $m_d = 2$, $m_e = 2$, $m_f = 2$, $m_g = 3$, $m_h = 3$, $m_i = 1$, $m_j = 2$, $m_k = 2$, $m_l = 3$, $m_m = 2$. The denoted solution is SO.

We have path costs for the used paths in the SO solution

$$\begin{aligned} &C_1^{ad} = 2, \quad C_1^{af} = 3, \quad C_1^{ag} = 5,\\ &C_2^{adi} = 3, \quad C_2^{afj} = 4,\\ &C_3^{g} = 2,\\ &C_4^{ek} = 5, \quad C_4^{hm} = 7. \end{aligned}$$

So the order costs are

$$C_1 = 2 + 3 + 5 = 10, \quad C_2 = 3 + 4 = 7, \quad C_3 = 2 \cdot 2 = 4 \text{ and } C_4 = 2 \cdot 5 + 2 \cdot 7 = 24,$$

while the cheapest path costs are

$$k_1 = 2 \cdot 2 + 3 = 7, \quad k_2 = 3 + 4 = 7, \quad k_3 = 2 \cdot 2 = 4 \text{ and } k_4 = 3 + 2 \cdot 4 + 5 = 16.$$

Algorithm 4 gives two connected components: $\mathcal{V}_1 = \{1, 2, 4\}$ and $\mathcal{V}_2 = \{3\}$.
When looking at the paths used in the SO, the question arises, why is order 3 in a separate connected component, while it looks like its taking its cheapest path at the expense of order 4 (it looks like arc g is a bottleneck arc for orders 3 and 4)? In this network however, we can see that it makes no difference for the two last arriving containers of order 4, if they are transported via path gl or hm. So order 3 does not influence the order costs of orders 1,2 and 4 and therefore is not connected to them.
With the components $\mathcal{V}_1$ and $\mathcal{V}_2$, solving the NP-β gives us order tolls

$$\beta_1 = -\frac{13}{30}, \quad \beta_2 = 2\frac{17}{30} \text{ and } \beta_4 = -2\frac{2}{15},$$

which give the tolled order costs

$$C_{\beta 1} = 9\frac{17}{30}, \quad C_{\beta 2} = 9\frac{17}{30}, \quad C_{\beta 3} = 4 \text{ and } C_{\beta 4} = 21\frac{13}{15}.$$

We obtain ratios $r_{\beta 1} = r_{\beta 2} = r_{\beta 4} = 1\frac{11}{30}$ and $r_{\beta 3} = 1$. △

5.6. Solutions of toll on orders

After finding the connected components, the question arises whether every NP-β of Algorithm 3 has a (unique) solution? For each component $\mathcal{V}_s$, there are $\frac{1}{2}|\mathcal{V}_s|(|\mathcal{V}_s|-1)$ terms in the objective, and we can state that the best solution possible has objective value 0, so then each term has to be equal to zero.

$$0=\left|\frac{C_i+\beta_i}{k_i}-\frac{C_j+\beta_j}{k_j}\right|=\left|\frac{k_j(C_i+\beta_i)-k_i(C_j+\beta_j)}{k_ik_j}\right| \Leftrightarrow k_i\beta_j-k_j\beta_i=k_jC_i-k_iC_j. \tag{5.7}$$

We can make a generalization of the optimal solution for tolls on orders: Given we have a problem with m orders, we can create the linear system $A\boldsymbol{\beta}=\boldsymbol{b}$ with A a block diagonal matrix:

$$A=\begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_m \end{bmatrix}$$

So A consists of m block matrices, one for each component $\mathcal{V}_s=\{1,\dots,n\}\ \forall\ 1\le s\le m$, with $n=|\mathcal{V}_s|$ the number of orders in $\mathcal{V}_s$:

$$A_i=\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ \mathbf{1}_{1,n} \end{bmatrix} \text{ with } D_j=\begin{bmatrix} \mathbf{0}_{1,j-1} & k_{j+1} & -k_j & 0 & \cdots & 0 \\ \mathbf{0}_{1,j-1} & k_{j+2} & 0 & -k_j & \ddots & \vdots \\ \mathbf{0}_{1,j-1} & \vdots & \vdots & \ddots & \ddots & 0 \\ \mathbf{0}_{1,j-1} & k_n & 0 & \cdots & 0 & -k_j \end{bmatrix},\ 1\le j\le n-1, \tag{5.8}$$

and $\mathbf{1}_{1,n}$ is a row vector of n ones, $\mathbf{0}_{1,j-1}$ a row vector of $j-1$ zeros and $\mathbf{0}_{1,0}:=\emptyset$.

The vector $\boldsymbol{b}=\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_m \end{bmatrix}$ consists of vectors $\boldsymbol{b}_s$ for each component $\mathcal{V}_s=\{1,\dots,n\}\ \forall\ 1\le s\le m$ with $n=|\mathcal{V}_s|$:

$$\boldsymbol{b}_i=\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_{n-1} \\ 0 \end{bmatrix} \text{ with } \boldsymbol{z}_j=\begin{bmatrix} k_jC_{j+1}-k_{j+1}C_j \\ k_jC_{j+2}-k_{j+2}C_j \\ \vdots \\ k_jC_n-k_nC_j \end{bmatrix} \tag{5.9}$$

A is a $\sum_{s=1}^{m}\left(\frac{1}{2}|\mathcal{V}_s|\cdot(|\mathcal{V}_s|-1)+1\right)\times|\mathcal{W}|$ matrix and $\boldsymbol{b}$ has length $\sum_{s=1}^{m}\left(\frac{1}{2}|\mathcal{V}_s|\cdot(|\mathcal{V}_s|-1)+1\right)$.
To draw some conclusions out of this equality $A\boldsymbol{\beta}=\boldsymbol{b}$, we first introduce some definitions from Linear Algebra:

Definition 6 (Row-echelon form)**.** A matrix is in row-echelon form if:

- All zero rows have been moved to the bottom.
- The leading nonzero element (also called a pivot) in any row is farther to the right than the leading nonzero element in the row just above it.
- In each column containing a leading nonzero element, the entries below that leading nonzero element are 0.

We can apply elementary row operations (swapping positions of rows, multiply a row by a nonzero scalar, add one row to a scalar multiple of another row) to modify the matrix until we obtain a row-echelon form.

Theorem 4. *[11, Theorem 1.7] Let $A\boldsymbol{x} = \boldsymbol{b}$ be a linear system, and let $[A|\boldsymbol{b}] \sim [H|\boldsymbol{c}]$, where H is in row-echelon form.*

1. *The system $A\boldsymbol{x} = \boldsymbol{b}$ is inconsistent if and only if the augmented matrix $[H|\boldsymbol{c}]$ has a row with all entries 0 to the left of the partition and a nonzero entry to the right of the partition.*

2. *If $A\boldsymbol{x} = \boldsymbol{b}$ is consistent and every column of H contains a pivot, the system has a unique solution.*

3. *If $A\boldsymbol{x} = \boldsymbol{b}$ is consistent and some column of H has no pivot, the system has infinitely many solutions, with as many free variables as there are pivot-free columns in H.*

We can thus reduce our problem $A\boldsymbol{\beta} = \boldsymbol{b}$ to $H\boldsymbol{\beta} = \boldsymbol{c}$, where H is in row-echelon form. When solving an NP-β with two orders and one connected component $\mathcal{V}_1$, we can easily verify that there always exists a unique solution by reducing the initial system:

$$\left[\begin{array}{cc|c} k_2 & -k_1 & k_1C_2 - k_2C_1 \\ 1 & 1 & 0 \end{array}\right] \sim \textit{Multiply row 2 by } k_2 \textit{ and subtract row 1 from row 2.}$$
$$\left[\begin{array}{cc|c} k_2 & -k_1 & k_1C_2 - k_2C_1 \\ 0 & k_1 + k_2 & k_2C_1 - k_1C_2 \end{array}\right]$$

We can see that Statement 2 of Theorem 4 holds, because $k_w > 0 \ \forall \ w \in \mathcal{W}$.

Then the solution is

$$\beta_2 = \frac{k_2C_1 - k_1C_2}{k_1 + k_2},$$
$$\beta_1 = \frac{k_1C_2 - k_2C_1}{k_2} + \frac{k_1}{k_2}\beta_2 = \frac{k_1C_2 - k_2C_1}{k_2} + \frac{k_1\left(k_2C_1 - k_1C_2\right)}{k_2\left(k_1 + k_2\right)} = \frac{k_1C_2 - k_2C_1}{k_1 + k_2}.$$

We only need to show that the constraints $\beta_w \geq -C_w$ are satisfied for all $w \in \mathcal{W}$:

$$-C_2 \leq \beta_2 = \frac{k_2C_1 - k_1C_2}{k_1 + k_2}$$
$$-C_1 \leq \beta_1 = \frac{k_1C_2 - k_2C_1}{k_1 + k_2}$$
$$\Leftrightarrow$$
$$0 \leq \frac{k_1\left(C_1 + C_2\right)}{k_1 + k_2}$$
$$0 \leq \frac{k_2\left(C_1 + C_2\right)}{k_1 + k_2}$$

So we see that the constraints are always satisfied, since $k_1,\ k_2,\ C_1,\ C_2 \geq 0$.

We can generalize this to a problem with n orders with m components: Suppose we have the NP-β problem with n orders, and $\mathcal{V}_i = \{1, \dots, n\}$, so $|\mathcal{V}_i| = n$ for all $1 \leq i \leq m$. We have the problem $A\boldsymbol{\beta} = \boldsymbol{b}$, with A being a block matrix. We will show there exist a unique solution for each sub-problem $A_i\boldsymbol{\beta} = \boldsymbol{b}_i$, and so a unique solution exist for the original problem.
The row-echelon form consist of the first row of each matrix D_j, so that are the rows $\begin{bmatrix}\mathbf{0}_{1,j-1} & k_{j+1} & -k_j & \mathbf{0}_{1,n-(j+1)}\end{bmatrix} \ \forall \ 1 \leq j \leq n-1$. All other rows in the matrices D_j can be written as a linear combination of rows $\begin{bmatrix}\mathbf{0}_{1,j-1} & k_{j+1} & -k_j & \mathbf{0}_{1,n-(j+1)}\end{bmatrix}$, so those rows are equal to a zero row in the row-echelon form.

For block matrix A_i and corresponding vector $\boldsymbol{b}_i$, we obtain the following row-echelon form $[H_i|\boldsymbol{c}_i]$:

$$H_i = \begin{bmatrix} k_2 & -k_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & k_3 & -k_2 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & k_4 & -k_3 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 & k_{n-1} & -k_{n-2} & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & k_n & -k_{n-1} \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & -\sum_{i=1}^{n} k_i \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{c}_i = \begin{bmatrix} k_1C_2 - k_2C_1 \\ k_2C_3 - k_3C_2 \\ k_3C_4 - k_4C_3 \\ \vdots \\ k_{n-2}C_{n-1} - k_{n-1}C_{n-2} \\ k_{n-1}C_n - k_nC_{n-1} \\ \sum_{j=1}^{n-1} k_jC_n - \sum_{j=1}^{n-1} k_nC_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

So we can conclude with Theorem 4.2 that for each toll problem with one connected component, there is a unique toll solution, because each column in H_i contains a pivot. Then, it follows that each row in block diagonal matrix H has a pivot in each column, and for the zero rows, $\boldsymbol{c}$ contain a zero in that row number, so together with the inequality constraints $\beta_w \geq -C_w \; \forall \; w \in \mathcal{W}$ satisfied, Theorem 4.2 holds. We then can conclude that if a solution exists for a toll order problem given the inequality constraints, the equality constraints provide a unique solution.

6

Path tolls based on order fairness

In Chapter 4 we have seen how we can adjust path tolls in order to create new tolled costs for the STN, in which the customers can choose their own traveling paths, and the path costs are constructed in such a way that they will choose paths that minimize the total cost of the network (SO) and are the cheapest paths for themselves (UE). In Chapter 5 we have seen how we can redivide the made costs over all orders, such that the use of the network is payed by all customers, and no customer is more harmed when regarding costs than others when they are using the same part of the network.
When adding tolls to paths as in Chapter 4, it may feel like we are misleading the customers, because the tolled path costs can differ a lot from the initial path costs. when taking the costs of the original networks into account. To avoid this misleading of customers, we can couple the constraints of fairness as used in the objective function of order tolls, to make the path based solutions a more fair User Equilibrium, by divide the costs over all orders in the network, instead of assign higher path costs to certain orders only. In this Chapter we will explain our procedure.
We assume the customers are unfamiliar with the initial path costs, like we assumed in Chapter 4.

Recall the SO problem (4.4) of Algorithm 2 on page 42. To find path tolls, we start solving the SO problem:

$$
\begin{aligned}
\min \; & \sum_{p \in \mathcal{P}} C_w^p f_p \\
\text{s.t.} \; & x_a = \sum_{p \in \mathcal{P}} \delta_{ap} f_p \;\; \forall \; a \in \mathcal{A} \\
& \sum_{p \in \mathcal{P}_w} f_p = d_w \;\; \forall \; w \in \mathcal{W} \\
& x_a \leq m_a \;\; \forall \; a \in \mathcal{A} \\
& f_p \in \mathbb{N}_0 \;\; \forall \; p \in \mathcal{P} \\
& x_a \in \mathbb{N}_0 \;\; \forall \; a \in \mathcal{A}
\end{aligned}
$$

With the SO solution found, the next step is calculating the path tolls. Therefore we use the NP-β (4.5)-(4.8) from Algorithm 2, but with the extra constraints:

$$
r_{\beta w_1} = r_{\beta w_2} \iff \frac{C_{w_1} + \beta_{w_1}}{k_{w_1}} = \frac{C_{w_2} + \beta_{w_2}}{k_{w_2}} \iff
$$
$$
\frac{\sum_{p \in \mathcal{P}_{w_1}} \left(C_{w_1}^p + \beta_{w_1}^p \right) f_p}{k_{w_1}} = \frac{\sum_{p \in \mathcal{P}_{w_2}} \left(C_{w_2}^p + \beta_{w_2}^p \right) f_p}{k_{w_2}} \quad \forall \; w_1, w_2 \in \mathcal{V}_s, \; \forall \; s \in \{1, \dots, k\},
$$

with order sets $\mathcal{V}_s$ as in Algorithm 4 on page 76.
Then we obtain the following NP-β for paths tolls:

$$\min \sum_{w\in\mathcal{W}} \sum_{p\in h_{out,w}} |\beta_w^p| \tag{6.1}$$

$$\text{s.t.} \sum_{w\in\mathcal{W}} \sum_{p\in h_{in,w}} \beta_w^p \underline{f_p} = 0 \tag{6.2}$$

$$\sum_{w\in\mathcal{V}_s} \sum_{p\in h_{in,w}} \beta_w^p \underline{f_p} = 0 \;\forall\; s \in \{1,\ldots,k\} \tag{6.3}$$

$$\frac{\sum_{p\in\mathcal{P}_{w_1}} \left(C_{w_1}^p + \beta_{w_1}^p\right) \underline{f_p}}{k_{w_1}} = \frac{\sum_{p\in\mathcal{P}_{w_2}} \left(C_{w_2}^p + \beta_{w_2}^p\right) \underline{f_p}}{k_{w_2}} \;\forall\; w_1, w_2 \in \mathcal{V}_s,\; \forall\; s \in \{1,\ldots,k\} \tag{6.4}$$

$$\beta_w^i - \beta_w^j \leq C_w^j - C_w^i \;\forall\; (i,j),\; i \in h_{in,w},\; j \in h_{out,w} \;\forall\; w \in \mathcal{W} \tag{6.5}$$

$$\beta_w^p \geq -C_w^p \;\forall\; p \in \mathcal{P} \tag{6.6}$$

where the objective (6.1) minimizes the tolls added to unused paths, Constraint 6.2 is enforces the tolls on used paths to sum up to zero, Constraint 6.3 enforces the same for each connected component $\mathcal{V}_s$. Constraint 6.4 ensures the ratio of all orders in one connected component are equal, Constraint 6.5 makes sure the path cost of chosen paths is cheaper than or equal to the unused path costs for each order, and Constraint 6.6 ensures that no negative path tolls arise.
In comparison to the NP-β as used in Chapter 4, Constraint 6.4 is added.

In the following Sections 6.1 up and until 6.4 we will show several examples for the different cases as desribed in 3.2, and in Section 6.5 we will show there always exists solutions for the NP-β, so there always is a toll solution to achieve both a UE and an SO.

6.1. Case 1

Case 1 is the situation in which $d_w = 1 \;\forall\; w \in \mathcal{W}$, and $m_a = 1 \;\forall$ arcs between different locations. For all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
To show the what toll solutions we obtain, we use the STN as in Example 10:

Example 28. The SO solution is given in Figure 6.1.

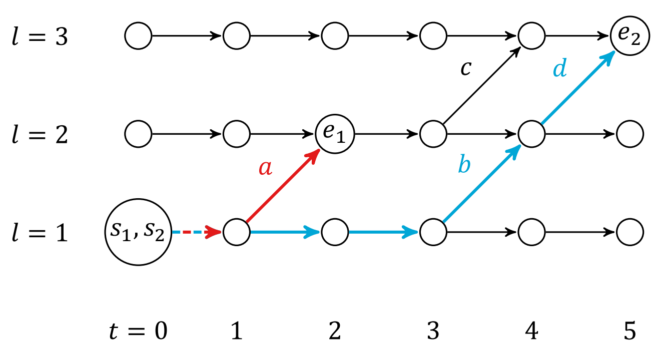


Figure 6.1: STN with two orders, $d_1 = d_2 = 1$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

In this SO solution the path costs are $C_1^a = 2$, $C_1^b = 4$, $C_2^{ac} = 4$, $C_2^{ad} = 5$, $C_2^{bd} = 5$. The chosen paths are $h_{in,1} = \{a\}$, $h_{in,2} = \{bd\}$, so the costs per customer are $C_1 = C_1^a = 2$, $C_2 = C_2^{bd} = 5$, while the cheapest path costs are $k_1 = 2$, $k_2 = 4$. The output of the NP-β is denoted below:

$$\beta_1^a = \frac{1}{3}, \quad \beta_2^{ac} = \frac{2}{3}, \quad \beta_2^{bd} = -\frac{1}{3},$$
$$C_{\beta 1}^a = 2\frac{1}{3}, \quad C_{\beta 2}^{ac} = 4\frac{2}{3}, \quad C_{\beta 2}^{bd} = 4\frac{2}{3}.$$

The total order costs are:

$$C_{\beta 1} = C_{\beta 1}^a = 2\frac{1}{3}, \quad C_{\beta 2} = C_{\beta 2}^{bd} = 4\frac{2}{3}.$$

If we solve the NP-β of order toll, we obtain the same tolled order costs as when solving the NP-β described in this Chapter. Here the ratio is $r_{\beta w} = 1\frac{1}{6}$ for both customers.
We can compare this solution to the path toll solution as found in Example 10. Here we found path tolls $\beta_1^a = 1$ and $\beta_2^{bd} = -1$, which gave the tolled ratios $r_{\beta 1} = \frac{3}{2}$ and $r_{\beta 2} = 1$. With those tolls order 1 had to pay the extra costs made in the network. With the path tolls based on order fairness, the extra costs are divided over both orders. $\triangle$

6.2. Case 2

Case 2 is the situation in which $d_w \geq 1 \; \forall \; w \in \mathcal{W}$, and $m_a = 1 \; \forall$ arcs between different locations. or all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
We show an example for Case 2: we use the STN as in Example 15:

Example 29. The SO solution is given in Figure 6.2.

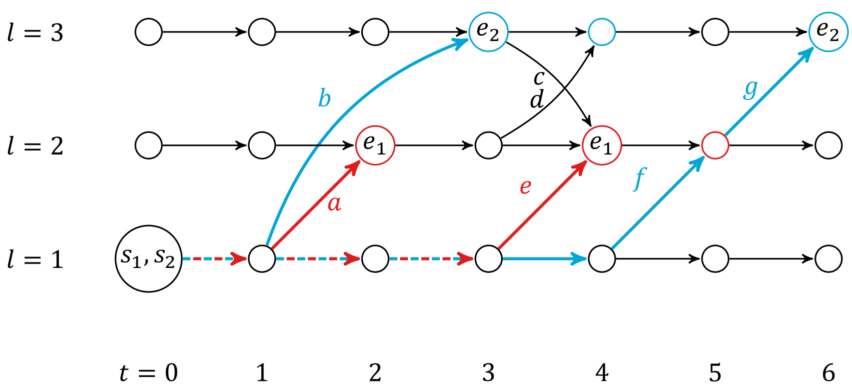


Figure 6.2: STN with two orders, $d_1 = d_2 = 2$, with $m_{a_i} = 1$ for all arcs between two different locations, $m_{a_i} = \infty$ otherwise. The denoted solution is SO.

In this SO solution the path costs are $C_1^a = 2$, $C_1^e = 4$, $C_1^{bc} = 3$, $C_1^f = 5$, $C_2^b = 3$, $C_2^{ad} = 4$, $C_2^{ag} = 6$, $C_2^{eg} = 6$, $C_2^{fg} = 6$. The chosen paths are $h_{in,1} = \{a, e\}$, $h_{in,2} = \{b, fg\}$, so the costs per customer are $C_1 = C_1^a + C_1^e = 6$ and $C_2 = C_2^b + C_2^{fg} = 9$, while the cheapest path costs are $k_1 = 6$ and $k_2 = 7$. Again we calculate the NP-β for path tolls and obtain the tolls and tolled path costs:

$$\beta_1^a = 2, \quad \beta_1^e = -1\frac{1}{13}, \quad \beta_2^b = 1\frac{1}{26}, \quad \beta_2^{ad} = \frac{1}{26}, \quad \beta_2^{fg} = -1\frac{25}{26},$$
$$C_{\beta 1}^a = 4, \quad C_{\beta 1}^e = 2\frac{12}{13}, \quad C_{\beta 2}^b = 4\frac{1}{26}, \quad C_{\beta 2}^{ad} = 4\frac{1}{26}, \quad C_{\beta 2}^{fg} = 4\frac{1}{26}.$$

The total orders costs are:

$$C_1 = C_1^a + C_1^e = 6\frac{12}{13}, \quad C_2 = C_2^b + C_2^{fg} = 8\frac{1}{13}.$$

Here the ratios are $r_1 = r_2 = 1\frac{2}{13}$, so equal for both customers. △

We see that the addition of the ratio constraint provides a more fair division of the costs over the customers, given that they can decide between the path (and corresponding tolled path costs) we offered them.

6.3. Case 3

Case 3 is the situation in which $d_w = 1 \; \forall \; w \in \mathcal{W}$, and $m_a \geq 1 \; \forall$ arcs between different locations. or all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.

Example 30. In this example we have 5 orders all with demand 1: Order 1,2, 3 and for start at location 1, order 5 starts at location 2. Order 1 and 2 end at location 2 and the others at location 3. The SO solution is given in Figure 6.3, with s_w and e_w denoting the start end point of order w, respectively. All traveling arcs have capacity 1, except for arcs b and f, which have capacity $m_a = m_e = 2$.
So we get the following path sets per order with corresponding capacities:

$$\mathcal{P}_{cap,1} = \mathcal{P}_{cap,2} = \begin{Bmatrix} a, & b, & c \\ 2, & 1, & 1 \end{Bmatrix}$$

$$\mathcal{P}_{cap,3} = \mathcal{P}_{cap,4} = \begin{Bmatrix} ae, & af, & ag, & bf, & bg \\ 2, & 1, & 1, & 1, & 1 \end{Bmatrix}$$

$$\mathcal{P}_{cap,5} = \begin{Bmatrix} d, & e, & f, & g \\ 1, & 2, & 1, & 1 \end{Bmatrix}$$

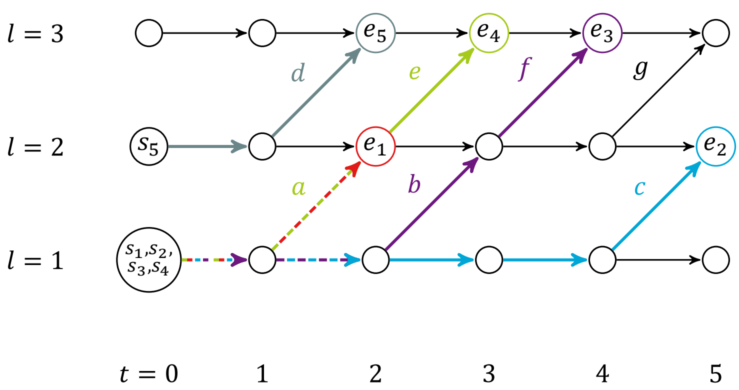


Figure 6.3: TN with five orders,$d_w = 1 \; \forall \; w \in \mathcal{W}$, with $m_a = m_e = 2$, $m_{a_i} = 1$ for $a_i \in \mathcal{A} \setminus \{a, e\}$, $m_{a_i} = \infty$ on waiting arcs. The denoted solution is SO.

We have costs

$$\begin{array}{lllll} C_{1,2}^a = 2, & C_{1,2}^b = 3, & C_{1,2}^c = 5, & & \\ C_{3,4}^{ae} = 3, & C_{3,4}^{af} = 4, & C_{3,4}^{ag} = 5, & C_{3,4}^{bf} = 4, & C_{3,4}^{bg} = 5, \\ C_5^d = 2, & C_5^e = 3, & C_5^f = 4, & C_5^g = 5, & \end{array}$$

where C_{w_1,w_2}^p denotes that for both orders the same path can be traversed, because those orders have the same origin and destination location.

The path sets follow from the SO solution:

$$h_{in,1} = \begin{Bmatrix} a \\ 1 \end{Bmatrix}, \qquad h_{out,1} = \begin{Bmatrix} a, & b, & c \\ 1, & 1, & 1 \end{Bmatrix},$$
$$h_{in,2} = \begin{Bmatrix} c \\ 1 \end{Bmatrix}, \qquad h_{out,2} = \begin{Bmatrix} a, & b \\ 2, & 1 \end{Bmatrix},$$
$$h_{in,3} = \begin{Bmatrix} ae \\ 1 \end{Bmatrix}, \qquad h_{out,3} = \begin{Bmatrix} ae, & af, & ag, & bg \\ 1, & 1, & 1, & 1 \end{Bmatrix},$$
$$h_{in,4} = \begin{Bmatrix} bf \\ 1 \end{Bmatrix}, \qquad h_{out,4} = \begin{Bmatrix} ae, & af, & ag, & bf, & bg \\ 2, & 1, & 1, & 1, & 1 \end{Bmatrix},$$
$$h_{in,5} = \begin{Bmatrix} d \\ 1 \end{Bmatrix}, \qquad h_{out,5} = \begin{Bmatrix} e, & f, & g \\ 2, & 1, & 1 \end{Bmatrix}.$$

We have two order sets: $\mathcal{V}_1 = \{1, 2, 3, 4\}$ and $\mathcal{V}_2 = \{5\}$. Solving the NP-β gives us

$$\beta_1^a = \frac{4}{5},\ \beta_2^c = -2\frac{1}{5},\ \beta_3^{bf} = \frac{1}{5},\ \beta_4^{ae} = 1\frac{1}{5},$$

where for orders 3 and 4 tolls are added to paths not in the SO solution. Then the adjusted path costs become

$$C_{\beta 1}^a = 1 + \frac{4}{5} = 2\frac{4}{5}, \quad C_{\beta 2}^c = 5 - 2\frac{1}{5} = 2\frac{4}{5}, \quad C_{\beta 3}^{bf} = 4 + \frac{1}{5} = 4\frac{1}{5}, \quad C_{\beta 4}^{ae} = 3 + 1\frac{1}{5} = 4\frac{1}{5}.$$

Now all orders have ratio $r_{\beta w} = 1\frac{2}{5}$. △

6.4. Case 4

Case 4 is the situation in which $d_w \geq 1\ \forall\ w \in \mathcal{W}$, and $m_a \geq 1\ \forall$ arcs between different locations. or all horizontal arcs within one location, the so-called waiting arcs, we have $m_a = \infty$.
We use the STN as in Example 27:

Example 31. We use the STN as in Figure 6.4.

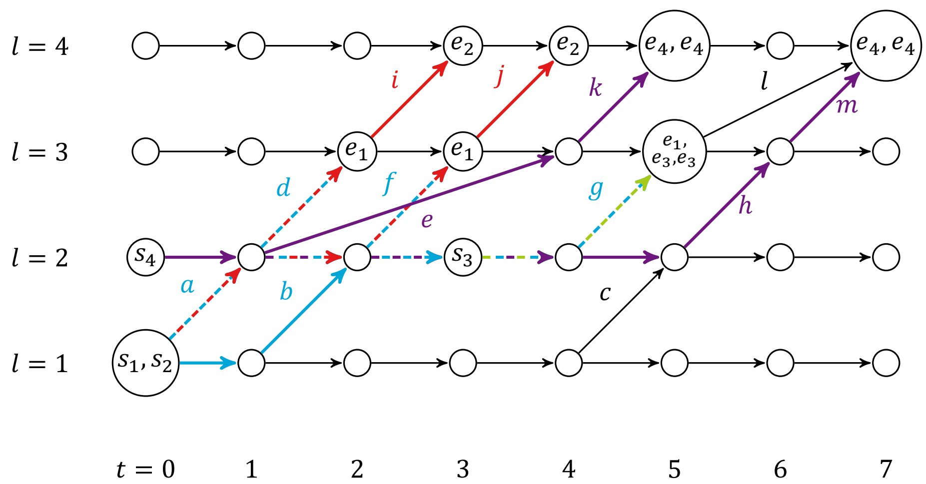


Figure 6.4: STN with four orders, $d_1 = 3$, $d_2 = d_3 = 2$, $d_4 = 4$, Capacities $m_a = 4$, $m_b = 4$, $m_c = 2$, $m_d = 2$, $m_e = 2$, $m_f = 2$, $m_g = 3$, $m_h = 3$, $m_i = 1$, $m_j = 2$, $m_k = 2$, $m_l = 3$, $m_m = 2$. The denoted solution is SO.

There are 5 orders, with demands $d_1 = 3$, $d_2 = 2$, $d_3 = 2$ and $d_4 = 4$. All diagonal arcs have capacity constraints: $m_a = 4$, $m_b = 4$, $m_c = 2$, $m_d = 2$, $m_e = 2$, $m_f = 2$, $m_g = 3$, $m_h = 3$, $m_i = 1$, $m_j = 2$, $m_k = 2$, $m_l = 3$ and $m_m = 2$.
The SO solution is given in Figure 6.4, with s_w and e_w denoting the start and end point for order w, respectively. Note that order 3 starts at $t = 3$, location $l = 2$.

We have path sets:

$$\mathcal{P}_{cap,1} = \begin{Bmatrix} ad, & ae, & af, & ag, & ah, & bf, & bg, & bh, & ch \\ 2, & 2, & 2, & 3, & 3, & f, & 3, & 3, & 2 \end{Bmatrix},$$

$$\mathcal{P}_{cap,2} = \begin{Bmatrix} adi, & adj, & adk, & adl, & adm, & aek, & ael, & aem, & afj, & afk, & afl, & afm, & agl, & agm, & ahm \\ 1, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 3, & 2, & 2 \end{Bmatrix},$$

$$\mathcal{P}_{cap,3} = \begin{Bmatrix} g, & h \\ 3, & 3 \end{Bmatrix},$$

$$\mathcal{P}_{cap,4} = \begin{Bmatrix} di, & dj, & dk, & dl, & dm, & ek, & el, & em, & fj, & fk, & fl, & fm, & gl, & gm, & hm \\ 1, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 2, & 3, & 2, & 2 \end{Bmatrix}.$$

We have path costs for the used paths in the SO solution

$$\begin{aligned} &C_1^{ad} = 2, \quad C_1^{af} = 3, \quad C_1^{ag} = 5, \\ &C_2^{adi} = 3, \quad C_2^{afj} = 4, \\ &C_3^{g} = 2, \\ &C_4^{ek} = 5, \quad C_4^{hm} = 7. \end{aligned}$$

So the order costs are $C_1 = 10$, $C_2 = 7$, $C_3 = 4$ and $C_4 = 24$, while the cheapest path costs are $k_1 = 7$, $k_2 = 7$, $k_3 = 4$ and $k_4 = 16$. Algorithm 4 gives two connected components: $\mathcal{V}_1 = \{1, 2, 4\}$ and $\mathcal{V}_2 = \{3\}$. △

6.5. Solutions of fairness tolls

The question arises whether we can always obtain a combined toll solution? For the inequality constraints we can always find a valid solution:

$$\beta_w^i \geq -C_w^i \;\; \forall \; i \in \mathcal{P}_w$$
$$\beta_w^i - \beta_w^j \leq C_w^j - C_w^i \;\; \forall \; (i,j), \; i \in h_{in,w}, \; j \in h_{out,w}$$

The first set of constraints provide a lower bound for all path tolls, and the last set of constraints give an upper bound for all paths tolls for paths $p \in \bigcup_{w \in \mathcal{W}} h_{in,w}$, the path tolls for the other paths are unbounded. We can always find a toll vector $\boldsymbol{\beta}$ that satisfies those inequality constraints. If $C_w^j - C_w^i < 0$ for some (i,j), $i \in h_{in,w}$, $j \in h_{out,w}$, set $\beta_w^j := C_w^i - C_w^j$ for all those paths j and $\beta_w^p := 0$ for all other paths. This shows the solution space is non-empty.

We now will investigate if the equality constraints provide a valid solution in combination with the inequality constraints. The equality constraints are

$$\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p = 0 \tag{6.7}$$

$$\sum_{w \in \mathcal{V}_s} \sum_{p \in h_{in,w}} \beta_w^p = 0 \;\; \forall \; s \in \{1, \dots, k\} \tag{6.8}$$

$$\frac{\sum_{p \in \mathcal{P}_{w_1}} \left(C_{w_1}^p + \beta_{w_1}^p \right) \underline{\underline{f_p}}}{k_{w_1}} = \frac{\sum_{p \in \mathcal{P}_{w_2}} \left(C_{w_2}^p + \beta_{w_2}^p \right) \underline{\underline{f_p}}}{k_{w_2}} \;\; \forall \; w_1, w_2 \in \mathcal{V}_s, \;\; \forall \; s \in \{1, \dots, k\}. \tag{6.9}$$

Constraint (6.7) is superfluous, because it is equal to summing up Constraints (6.8):

$$\sum_{w \in \mathcal{W}} \sum_{p \in h_{in,w}} \beta_w^p = 0 \;\Leftrightarrow\; \sum_{s=1}^{k} \sum_{w \in \mathcal{V}_s} \sum_{p \in h_{in,w}} \beta_w^p = 0.$$

If $d_w = 1\ \forall\ w \in \mathcal{W}$ this problem corresponds to finding solutions in Section 5.6. If $d_w \geq 1$ $\forall\ w \in \mathcal{W}$ we can rewrite constraint (6.9) for a pair of two orders i and j, $i,j \in \mathcal{W}$ (we assume $h_{in,i} = \{p_1, \dots, p_k\}$, $h_{in,j} = \{q_1, \dots, q_l\}$:

$$\frac{C_i + \beta_i}{k_i} = \frac{C_j + \beta_j}{k_j}$$
$$\Leftrightarrow\ k_j\beta_i - k_i\beta_j = k_iC_j - k_jC_i$$
$$\Leftrightarrow\ k_j \sum_{p\in\mathcal{P}_i} \beta_i^p \underline{f_p} - k_i \sum_{p\in\mathcal{P}_j} \beta_j^p \underline{f_p} = k_i \sum_{p\in\mathcal{P}_j} C_j^p \underline{f_p} - k_j \sum_{p\in\mathcal{P}_i} C_i^p \underline{f_p}.$$

We can calculate the tolls per connected component.

- If $\mathcal{V}_s = \{1\}$, then constraint (6.8) state:

$$\sum_{p\in h_{in,1}} \beta_1^p \underline{f_p} = 0.$$

  So we do not need to add tolls to the STN, because this order can take its cheapest paths, because it does not use any bottleneck arcs, which indicates there are no issues in traveling via its cheapest paths.

- If $\mathcal{V}_s = \{1,2\}$, then constraint (6.8):

$$\beta_i + \beta_j = 0\ \Leftrightarrow\ \beta_i = -\beta_j\ \Leftrightarrow\ \sum_{p\in\mathcal{P}_i} \beta_i^p \underline{f_p} = -\sum_{p\in\mathcal{P}_j} \beta_j^p \underline{f_p}.$$

  We can use this to continue our rewriting of constraint (6.9):

$$(k_i + k_j) \sum_{p\in\mathcal{P}_i} \beta_i^p \underline{f_p} = k_i \sum_{p\in\mathcal{P}_j} C_j^p \underline{f_p} - k_j \sum_{p\in\mathcal{P}_i} C_i^p \underline{f_p}$$
$$\Leftrightarrow\ \sum_{p\in\mathcal{P}_i} \beta_i^p \underline{f_p} = \frac{k_i \sum_{p\in\mathcal{P}_j} C_j^p \underline{f_p} - k_j \sum_{p\in\mathcal{P}_i} C_i^p \underline{f_p}}{k_i + k_j}.$$

  The equality always has a solution because $k_i + k_j \neq 0$.

- If $\mathcal{V}_s = \{1,2,3\}$, then (6.8):

$$\sum_{p\in\mathcal{P}_1} \beta_1^p \underline{f_p} + \sum_{p\in\mathcal{P}_2} \beta_1^p \underline{f_p} + \sum_{p\in\mathcal{P}_3} \beta_3^p \underline{f_p} = 0,$$

  and (6.9):

$$k_2 \sum_{p\in\mathcal{P}_1} \beta_1^p \underline{f_p} - k_1 \sum_{p\in\mathcal{P}_2} \beta_2^p \underline{f_p} = k_1 \sum_{p\in\mathcal{P}_2} C_2^p \underline{f_p} - k_2 \sum_{p\in\mathcal{P}_1} C_1^p \underline{f_p}$$
$$k_3 \sum_{p\in\mathcal{P}_1} \beta_1^p \underline{f_p} - k_1 \sum_{p\in\mathcal{P}_3} \beta_3^p \underline{f_p} = k_1 \sum_{p\in\mathcal{P}_3} C_3^p \underline{f_p} - k_3 \sum_{p\in\mathcal{P}_1} C_1^p \underline{f_p}$$
$$k_3 \sum_{p\in\mathcal{P}_2} \beta_2^p \underline{f_p} - k_2 \sum_{p\in\mathcal{P}_3} \beta_3^p \underline{f_p} = k_2 \sum_{p\in\mathcal{P}_3} C_3^p \underline{f_p} - k_3 \sum_{p\in\mathcal{P}_2} C_2^p \underline{f_p}$$

We see this problem corresponds to the order toll problem, where the constraints we just observed are contained in the objective function (see (5.7)), and so a solution exist for each problem (see Section 5.6). In this case there are infinitely many solutions, because the row-echelon form of the problem satisfies Theorem 4.3. We will show this, but first introduce some extra notation: with x_{j-1} we denote the number of paths for orders 1 up until $j-1$: $x_j = \left|\bigcup_{1\le i\le j} \mathcal{P}_i\right|$, and we denote path flow found in the SO solution by f_w^p instead of $\underline{f_p}$. We assume $l = |\mathcal{P}_j| \ \forall j \in \mathcal{W}$.

We have $A\boldsymbol{\beta} = \boldsymbol{b}$ with matrix $A = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_m \end{bmatrix}$, with sub-matrix $A_i = \begin{bmatrix} D_1 \\ \vdots \\ D_{n-1} \\ \boldsymbol{g}_{1,x_n} \end{bmatrix}$ for each $i \in \{1,\dots,m\}$ with $\mathcal{V}_i$ a connected component, $\mathcal{V}_i = \{1,\dots,n\}$. The matrices D_j, $1 \le j \le n-1$. are defined as follows:

$$D_j = \begin{bmatrix} \boldsymbol{0}_{1,x_{j-1}} & k_{j+1}f_j^{p_1} & \cdots & k_{j+1}f_j^{p_l} & -k_jf_{j+1}^{p_1} & \cdots & -k_jf_{j+1}^{p_l} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \boldsymbol{0}_{1,x_{j-1}} & k_{j+2}f_j^{p_1} & \cdots & k_{j+2}f_j^{p_l} & 0 & \cdots & 0 & -k_jf_{j+2}^{p_1} & \cdots & -k_jf_{j+2}^{p_l} & 0 & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \boldsymbol{0}_{1,x_{j-1}} & k_nf_j^{p_1} & \cdots & k_nf_j^{p_l} & 0 & \cdots & \cdots & \cdots & \cdots & 0 & -k_jf_n^{p_1} & \cdots & -k_jf_n^{p_l} \end{bmatrix},$$

and vector $\boldsymbol{g}_{1,x_n}$ consists of all path flows for orders the connected component i:

$$\boldsymbol{g}_{1,x_n} = \begin{bmatrix} f_1^{p_1} & \cdots f_1^{p_l} & f_2^{p_1} & \cdots f_2^{p_l} & \cdots & \cdots & \cdots & f_n^{p_1} & \cdots f_n^{p_l} \end{bmatrix}.$$

Vector $\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_m \end{bmatrix}$ consists of $\boldsymbol{b}_i = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_{n-1} \\ 0 \end{bmatrix}$ $\forall\ 1 \le i \le m$ with $\boldsymbol{z}_j = \begin{bmatrix} k_jC_{j+1} - k_{j+1}C_j \\ k_jC_{j+2} - k_{j+2}C_j \\ \vdots \\ k_jC_n - k_nC_j \end{bmatrix}$.

Now we want to show this system $A\boldsymbol{\beta} = \boldsymbol{b}$, has infinitely many solutions. We therefore need to show that in the row-echelon form $H\boldsymbol{\beta} = \boldsymbol{c}$, matrix H contains columns with no pivots. The row-echelon form of sub-matrix A_i is

$$H_i = \begin{bmatrix} k_2f_1^{p_1} & \cdots & k_2f_1^{p_l} & -k_1f_2^{p_1} & \cdots & -k_1f_2^{p_l} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & k_3f_2^{p_1} & \cdots & k_3f_2^{p_l} & -k_2f_3^{p_1} & \cdots & -k_2f_3^{p_l} & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & k_nf_{n-1}^{p_1} & \cdots & k_nf_{n-1}^{p_l} & -k_{n-1}f_n^{p_1} & \cdots & -k_{n-1}f_n^{p_l} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & -\sum_{j=1}^{n} k_jf_n^{p_1} & \cdots & -\sum_{j=1}^{n} k_jf_n^{p_l} \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

and the vector $\boldsymbol{c}_i = \begin{bmatrix} k_1C_2 - k_2C_1 \\ k_2C_3 - k_3C_2 \\ \vdots \\ k_{n-1}C_n - k_nC_{n-1} \\ \sum_{j=1}^{n-1} k_jC_n - \sum_{j=1}^{n-1} k_nC_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. We see that the number of pivots in H_i equals the number of orders, which is less than the number of path toll variables, so according to Theorem 4.3: if a solution exists, there are infinitely many solutions for this toll problem.

7

Conclusions and discussion

7.1. Conclusions

The goal of this thesis was to provide a method to obtain a User Equilibrium (UE) in a synchromodal Space Time Network (STN) in which we transport containers for multiple customers. We defined User Equilibrium as the solution where each customer can travel via its cheapest paths, and no customer is harmed by the route choice of other customers.
We expanded this goal to also finding a solution of assigning containers to modes where the solution is both System Optimal (SO) as well as User Equilibrium, by finding a System Optimal solution and add tolls to the orders or paths.

In most research about User Equilibria, transportation via roads is considered, where the speed of vehicles on roads is dependent of the occupancy of the road it travels. There the goal is to divide the vehicles over the network in such a way that for each vehicle in the network, the transportation time is minimized given the total network and its users.
In our synchromodal STN, the travel time of modalities is not dependent of the occupancy of water, rail and road. The STN consists of discrete time steps and locations to model the transportation of containers. We assumed the travel times of barges, trains and trucks are fixed and known beforehand and transportation schedules for those modalities are known as well. Our goal was to obtain a User Equilibrium in this deterministic network, by assigning containers to the different modalities, in order to deliver each container as fast as possible. Because all our modalities have fixed travel times, the travel times are not dependent of the occupancy of the transportation arcs in the network, however, travel times are dependent of the capacity constraints on the arcs.

In order to provide an SO solution in which a UE is reached as well, we applied tolls to the network costs. But in order to obtain a User Equilibrium, we need to define when a User Equilibrium is reached. For each type of tolls investigated we have a different definition of a UE. We considered three type of tolls: path based, order based and fair path based. For all type of tolls, we demand that tolls payed by customers have to sum up to zero, such that no profit is made out of those tolls.
For path tolls, we say a UE is reached when each customer can travel via its cheapest tolled paths, regardless of the path choices of other customers. For order tolls, we say a UE is reached when all extra costs (compared to the cheapest path costs) made in the network, are divided over the orders in a fair way, concerning the ratio of the payed costs compared to the cheapest path cost. For fair path tolls, we say a UE is reached when both customers can travel via their cheapest paths, and the extra costs in the network are divided in a fair way over the customers. In our approach, one unit of toll represents one unit of time.
The first step in all toll algorithms is to calculate the SO based on the path costs of containers traveling from their origin to their destination. The next step is to calculate tolls that are added to the path or order costs, depending on what kind of tolls we considered.

When considering order tolls, the tolled order costs (initial costs plus the tolls) are the prize customers have to pay to transport their goods. The order tolls provided a fair cost division: for each set of customers who want to use the same traveling arcs in the network, the ratio of their tolled order costs compared to their cheapest costs (those cheapest costs are calculated by assuming they are the only customer in the network) are equal. We showed there always exists a unique toll solution in which a UE is reached and the SO objective value is maintained.

When applying path based tolls, we assume customers do not know the path costs of the initial network (and thus also do not know their initial cheapest paths). Here the tolls are used to adjust the path costs, such that we can offer the customers a choice of tolled paths. Then when the customer gets assigned its cheapest tolled paths, those paths are in the SO solution and the solution is UE as well. The solution is UE because the offered path costs are the cheapest option according to the information available for the customer.

Because the path tolls described above are somewhat misleading, as we offer customers path costs different from the initial network path costs, we also provided a fair path toll. In this fair path toll we combined the fairness ratios of the order tolls (total payed costs compared to their cheapest path costs), but assume that customers choose their routes themselves. It is thus a combination of both path and order toll. When we calculate the tolled path costs here, they sum up to the tolled order costs as calculated in the previous toll case.

We succeeded in finding an approach to obtain both an SO and a UE solution on an STN. We showed for path tolls that given some properties of the path costs of the initial network, it is possible for path tolls to find a toll scheme where tolls are only added to paths which are used to transport containers. For order tolls we showed there always exists a unique toll solution and for fair path tolls there always exists infinitely many solutions.

Based on the facts that we want the UE solution to be fair to all customers and we want customers to choose their routes themselves, the fair path toll is the best type of toll to solve this problem. However, finding path tolls have to be done before the transportations, because we want customers to choose their routes themselves. So we need to have information of all incoming orders beforehand, in order to calculate which route choice suits the set of order best in order to obtain an SO and to create the UE tolled path costs. In practice however, often the demands of the orders arrive over time, so then with each new incoming order, one needs to recalculate the SO solution. When using path tolls, we also need to adjust the tolls on the paths each time a new order has been assigned to the network. So order based tolls are better applicable in practice, because although the SO solution needs to be recalculated with each incoming order, finding order based tolls only has to be done once (when all transportations are done). Then we only have to calculate the tolls once, while with path based tolls we need to recalculate the path tolls each time a new order has been assigned to the network.

It may be clear that this thesis is particularly a theoretical view of obtaining a User Equilibrium in a deterministic network. We showed a User Equilibrium can be obtained when using tolls to adjust the costs of the STN. The question is, how can we apply the principle of tolls on real life instances? In the literature a lot more is known about achieving a System Optimal solution in a transportation network. That is why we used an SO solution as a base for creating new path or order costs to provide a User Equilibrium while the total costs of the SO can be maintained. Our work is thus a contribution to the connection of SO to UE. We succeeded in finding a method to find of assigning containers to modes in a synchromodal STN, where the solution is both System Optimal as well as User Equilibrium. The method found for solving this problem is using tolls on the synchromodal STN.

7.2. Discussion and future work

Our work is mainly theoretical and we made a simplification of transportation networks in order to get an idea of how we could solve the UE problem. The main improvements of this research thus lie in applying our toll approach to real life instances. In order to do so, we still need to more research on some steps:

One important part of logistics planning is the due date of orders. In this thesis we did not take those due dates into account, we just wanted all orders to be delivered as fast as possible, in other words, as cheaply as possible. When we do take due dates into account, it can be the case that orders will arrive too late compared to this due date. We then need to add a penalty function to the cost objective function in order to minimize the number of orders arriving too late. With the tolls, it is possible to share the penalty costs by all orders who are causing the lateness of the delayed orders.

Another improvement of our work can be the application of Value Of Time (VOT), when regarding the fairness of solutions. In the transportation business, it is often the case that it is more important for some customers to be served on time that for others, in other words, some customers may allow lateness, while others will not. Then in order to keep customers as satisfied as possible, it is more important to avoid lateness (in the case due dates are taken into account) for customers who will not tolerate that. The VOT of those customers must be set to a high value, such that the assignment of their customers to modes weight more heavily.
The VOT can also be used to cover the problem in the case customers have different individual objectives. This can be for example when some customers want the containers to be transported as fast as possible, while other customers prefer the cheapest transport possible, regardless of the travel time of those transportations. The Value Of Times of customers can then be applied as weights in objective functions in order to satisfy all customers. Another option is to group the customers per objective type, and for those groups of orders create constraints based on their demands, while the objective can be minimizing the tolls used, as in our approach.

We worked on obtaining a UE solution is a deterministic network, so work has to be done in order to apply UE on problems with both flexible infrastructure and uncertainties, such as delays of modalities or delayed orders, and uncertain events.
Research can also be done concerning uncertain events (so dynamic incoming orders). As mentioned before, orders tolls can be applied on those uncertain events, but for path tolls is the first step to investigate how to calculate those path tolls. An option is to use historical data of container-to-route assignment.
When we also need to plan the modalities in the network (so the schedules of the transportations are not known in advance), it can be possible to take the User Equilibrium into account when making this planning. This can be done by creating the (tolled) cost schemes while making the planning of all modalities.

Bibliography

[1] Wardrop equilibria. http://dii.uchile.cl/~jcorrea/papers/Chapters/CS2010.pdf.

[2] How to take the dual of a linear program. http://www.cs.columbia.edu/coms6998-3/lpprimer.pdf. Accessed: 2017-12-11.

[3] Platform synchromodaliteit. http://www.synchromodaliteit.nl/definitie/. Accessed: 2018-06-14.

[4] Hillel Bar-Gera. *Origin-based Algorithms for Transportation Network Modeling*. PhD thesis, Chicago, IL, USA, 1999. AAI9954880.

[5] Hillel Bar-Gera, David Boyce, and Yu (Marco) Nie. User-equilibrium route flows and the condition of proportionality. *Transportation Research Part B: Methodological*, 46(3):440 – 462, 2012. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2011.10.010. URL http://www.sciencedirect.com/science/article/pii/S0191261511001639.

[6] Martin Beckmann, C.B. McGuire, and Christopher B. Winsten. *Studies in the Economics of Transportation*. Yale University Press in New Haven, 1956.

[7] Marlies Borchers, Paul Breeuwsma, Walter Kern, Jaap Slootbeek, Georg Still, and Wouter Tibben. Traffic user equilibrium and proportionality. *Transportation Research Part B*, 79: 149–160, 2015.

[8] Francesco Corman, Francesco Viti, and Negenborn Rudy R. Equilibrium models in multimodal container transport systems. *Springerlink.com*, 2015.

[9] Mohamed Didi-Biha, Patrice Marcotte, and Gilles Savard. *Path-based formulations of a bilevel toll setting problem*, pages 29–50. Springer US, Boston, MA, 2006. ISBN 978-0-387-34221-4. doi: 10.1007/0-387-34221-4_2. URL https://doi.org/10.1007/0-387-34221-4_2.

[10] Michael Florian and Donald Hearn. *Network Equilibrium and Pricing*, volume 23, pages 373–411. 01 2003. doi: 10.1007/0-306-48058-1_11.

[11] J.B. Fraleigh, R.A. Beauregard, and V.J. Katz. *Linear Algebra*. Number v. 1 in Featured Titles for Linear Algebra. Addison-Wesley, 1995. ISBN 9780201526752. URL https://books.google.nl/books?id=XRJ_QgAACAAJ.

[12] Deren Han and Hai Yang. The multi-class, multi-criterion traffic equilibrium and the efficiency of congestion pricing. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):753 – 773, 2008. ISSN 1366-5545. doi: https://doi.org/10.1016/j.tre.2007.07.011. URL http://www.sciencedirect.com/science/article/pii/S1366554507000889.

[13] D.W. Hearn and M.V. Ramana. Solving congestion toll pricing models. *Marcotte P., Nguyen S. (eds) Equilibrium and Advanced Transportation Modelling. Centre for Research on Transportation*, 1998.

[14] L. Jiang and H. Mahmassani. Toll pricing: Computational tests for capturing heterogeneity of user preferences. *Transportation Research Record: Journal of the Transportation Research Board*, page 105–115, 2013. doi: 10.3141/2343-13.

[15] Torbjörn Larsson and Michael Patriksson. Side constrained traffic equilibrium models - traffic management through link tolls. *Marcotte P., Nguyen S. (eds) Equilibrium and Advanced Transportation Modelling. Centre for Research on Transportation*, 1998.

[16] Nadav Levy, Ido Klein, and Eran Ben-Elia. Emergence of cooperation and a fair system optimum in road networks: A game-theoretic and agent-based modelling approach. *Research in Transportation Economic*, 2016.

[17] Shu Lu and Yu (Marco) Nie. Stability of user-equilibrium route flow solutions for the traffic assignment problem. *Transportation Research Part B: Methodological*, 44(4):609 – 617, 2010. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2009.09.003. URL http://www.sciencedirect.com/science/article/pii/S0191261509001155.

[18] Toshihiko Miyagi and Genaro C. Peque. Informed-user algorithms that converge to nash equilibrium in traffic games. *Procedia - Social and Behavioral Sciences*, 54:438 – 449, 2012. ISSN 1877-0428. doi: https://doi.org/10.1016/j.sbspro.2012.09.762. URL http://www.sciencedirect.com/science/article/pii/S1877042812042243. Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris.

[19] Srinivas Peeta and Hani S. Mahmassani. System optimal and user equilibrium time-dependent traffic assignment in congested networks. *Annals of Operations Research*, 1995.

[20] Bart Riessen, Rudy R. Negenborn, and Rommert Dekker. Synchromodal container transportation: An overview of current topics and research opportunities. *6th International Conference on Computational Logistics (ICCL'15)*, pages 386–397, 2013.

[21] Tim Roughgarden. Selfish routing and the price of anarchy. *OPTIMA-2007*, 2006. URL https://theory.stanford.edu/~tim/papers/optima.pdf.

[22] Martin Skutella. *An Introduction to Network Flows over Time*, pages 451–482. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-76796-1. doi: 10.1007/978-3-540-76796-1_21. URL https://doi.org/10.1007/978-3-540-76796-1_21.

[23] Kenneth A. Small. Using the revenues from congestion pricing. *Transportation*, 19(4): 359–381, Dec 1992. ISSN 1572-9435. doi: 10.1007/BF01098639. URL https://doi.org/10.1007/BF01098639.

[24] Mariska van Essen, Tom Thomas, Eric van Berkum, and Caspar Chorus. From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels. *Transport Reviews*, 36(4):527–548, 2016. doi: 10.1080/01441647.2015.1125399. URL https://doi.org/10.1080/01441647.2015.1125399.

[25] Nicolas Wagner. The dynamic user equilibrium on a transport network : mathematical properties and economic applications. 2014.

[26] C. Wang and Y. Tang. The discussion of system optimism and user equilibrium in traffic assignment with the perspective of game theory. *Transportation Research Procedia*, 25: 2974,2983, 2017. doi: 10.1016/j.trpro.2017.05.200.

[27] John Glen Wardrop. Some theoretical aspects of road traffic research. *Road Engineering Division Meeting*, Road Paper No. 36:325 – 378, 1952.

[28] Laurence A. Wolsey. *Integer Programming.* John Wiley & Sons, 1998.

[29] Wangtu Xu, Lixin Miao, and Wei-Hua Lin. Stochastic user equilibrium assignment in schedule-based transit networks with capacity constraints. *Discrete Dynamics in Nature and Society*, Vol. 2012, Article ID 910754, 2012. doi: https://doi.org/10.1155/2012/910754.

[30] Hai Yang and Hai-Jun Huang. Social and spatial equities and revenue redistribution. pages 203–238, 2005.

[31] Hai Yang and Hai-Jun Huang. Fundamentals of user-equilibrium problems. pages 13 – 46, 2005.

[32] Hai Yang and Xiaoning Zhang. Existence of anonymous link tolls for system optimum on networks with mixed equilibrium behaviors. *Transportation Research Part B: Methodological*, 42(2):99 – 112, 2008. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2007.07.001. URL http://www.sciencedirect.com/science/article/pii/S0191261507000719.