Flows over time with commodity-dependent transit times

Diploma Thesis

Department of Computer Science
Technische Universität Dortmund

Martin Groß
August 24, 2009

Advisers:
Prof. Dr. Martin Skutella
Combinatorial Optimization & Graph Algorithms
Department of Mathematics
Technische Universität Berlin

Prof. Dr. Petra Mutzel
Algorithm Engineering
Department of Computer Science
Technische Universität Dortmund

I would like to thank my advisers Prof. Martin Skutella and Prof. Petra Mutzel for their support and guidance which enabled me to write this thesis. Furthermore I am grateful to the members of the chair of Algorithm Engineering of the TU Dortmund and the Combinatorial Optimization Group of the TU Berlin for many interesting talks and discussions. Finally I would like to thank my family for supporting me throughout my studies and diploma thesis which I am deeply grateful for.

Contents

1 Introduction **1**

2 Definitions **5**

2.1 Network Flows 5

2.2 Common Problems 13

3 Known Results **21**

4 Models **27**

4.1 Commodity-Dependent Transit Times 27

4.2 Complexity 32

4.3 Overtaking and Transit-Time-Restrictions 39

5 Quickest Transshipments **41**

5.1 FPTAS 41

6 Earliest Arrival Transshipments **49**

6.1 Existence 49

6.2 Approximation 51

7 Conclusion **57**

List of Figures **59**

Bibliography **63**

Chapter 1

Introduction

Motivation and Background

Network flows are a useful tool for solving several classes of optimization problems such as, but not limited to, routing and transportation problems. These problems usually consist of an underlying network of nodes and arcs that model a real-life network (e.g. a network of roads) and functions on the nodes and arcs that are used to express problem-specific constraints. For the case of routing or transportation problems there are often supplies and demands on the nodes that specify an amount of goods or agents that must be routed or transported from the nodes with supplies (called *sources*) to the nodes with demands (called *sinks*). In the context of network flows, these goods or agents are represented by flow. Usually, there are capacities on the arcs limiting the amount of flow that can use an arc. Network flow models based on a network with capacities, supplies and demands are called *classical* (or sometimes *static*) network flows.

Classical network flows are powerful enough to serve as a foundation for solving optimization problems and have been well researched (see e.g. Ahuja, Magnati, Orlin [1]). For the case of classical network flows, many interesting flow problems are known to be efficiently solvable. Unfortunately, classical network flows have a significant disadvantage when used to model routing or transportation problems: the travelling involved in routing or transportation does often not happen instantaneously but over a certain amount of time. Therefore routing or transportation decisions being made in one part of the network have an effect on other parts of the network only after a certain delay. This *temporal dimension* of the problem cannot easily be modeled by classical network flows and usually cannot simply be ignored, either.

In order to solve this problem, *flows over time* (sometimes called *dynamic flows*) were introduced by Ford and Fulkerson (see [11]). A flow over time is basically a classical flow with additional transit times on the arcs specifying the amount of time necessary to traverse an arc. This results in a network flow model that includes a temporal dimension, rendering it powerful enough to model transportation and routing problems more realistically. As to be expected, this power comes at the price of increased complexity. While flow over time problems can be reduced to classical flow problems (as described by Ford and Fulkerson) the reduction technique (*time expansion*) employed leads to a blow-up of the underlying network that is only pseudopolynomial in the input size. For some flow over time problems there are efficient algorithms that do not use time expansion, however.

Unfortunately, even flows over time are sometimes lacking in accuracy when modeling real-life problems. This stems mostly from the fact that the transit time of flow over an arc is constant and the same for all units of flow. Therefore, it is neither possible to model vehicles moving at different speeds through the network nor can the transit times be dependent on the current traffic on an arc - which would be more realistic in some scenarios.

In order to model transit times that depend on the current traffic *flow-dependent* transit times have been proposed, see Köhler and Skutella [22] as well as Hall, Langkau and Skutella [18]. These models allow an even more realistic modeling of traffic but again at the cost of increased complexity. Flow units that are subject to different transit times can be considered in the context of *multicommodity flows*. A multicommodity flow consists of several normal network flows with each flow representing one of several commodities (different kinds of goods for example). To the knowledge of the author, there has been no research on multicommodity flows with *commodity-dependent* transit times, i.e. transit times that differ for each of the commodities. In this thesis, we will introduce models for commodity-dependent transit times, analyse the complexity of flow problems in this setting and develop algorithms for solving them.

Outline of this thesis

This thesis is divided into seven chapters, including this one.

Chapter 2. In the second chapter we define the formal foundations for this thesis, including the classical network flow models as well as models for flows over time. Furthermore, we specify network flow problems of interest to us and introduce useful concepts and tools for dealing with these problems, such as residual networks and time expansion.

Chapter 3. In the third chapter we take a look at existing results for the problems given in Chapter 1. Due to the lack of research on these problems in the setting of commodity-dependent transit times we focus on results in other, simpler settings.

Chapter 4. Here we consider several models for commodity-dependent transit times and analyse their advantages and disadvantages, especially the effect on the complexity of our problems in these models.

Chapter 5. In this chapter we develop a fully polynomial time approximation scheme for the multicommodity quickest transshipment problem with commodity-dependent transit times based on condensed time-expanded networks.

Chapter 6. This chapter is devoted to the analysis of earliest arrival transshipment problems. Although these problems are of great interest in the context of evacuation planning, it is usually not guaranteed that a solution to an earliest arrival transshipment problem exists. We analyse the existence of earliest arrival flows in our setting and consider approximations to earliest arrival transshipments in the case that earliest arrival transshipments do not exist. More specifically, we analyse the existence of approximations to earliest arrival transshipments and develop ways of computing them.

Chapter 7. In this chapter we conclude this thesis by giving possible avenues of further research.

Chapter 2

Definitions

In this chapter we introduce the foundations this thesis is built on on the assumption that the reader is familiar with basic concepts of complexity theory, linear programming and combinatorial optimization. An introduction to these topics can be found in textbooks by Ahuja, Magnati and Orlin [1], Korte and Vygen [23] and Schrijver [26, 27].

2.1 Network Flows

We begin by defining networks and related concepts which will serve as the foundation for our problems.

Definition 2.1.1 (Network)**.** *A* network *is a directed graph $G = (V, A)$ consisting of a set of nodes $V = \{v_1, \ldots, v_n\}$ and a set of arcs $A = \{e_1 = (v_{x_1}, v_{y_1}), \ldots, e_m = (v_{x_m}, v_{y_m})\}$ with $v_{x_i}, v_{y_i} \in V$ for $i = 1, \ldots, m$. Note that we consider arcs $e_i = (v, w), e_j = (v, w), i \neq j$ to be different. For an arc $e = (v, w)$ we call w the* head *of e and v the* tail *of e. We call arcs $e = (v, v) \in A$* loops *and we call arcs $e = (v, w), e' = (v, w)$* parallel*. Furthermore, we write $n := |V|$ and $m := |E|$ for the number of nodes and arcs, respectively.*

Note that our definition of networks does permit loops and parallel arcs. Unless stated otherwise, we will assume that our networks do not contain loops, however. We will see later in this section that this assumption is often of no consequence to the power of the models we are considering. In cases where no parallel edges exist we simplify our notations by identifying an arc $e = (v, w)$ with its pair of nodes (v, w).

Throughout this thesis, we will consider network problems and refer to their respective networks as $G = (V, A)$ as defined above. We will assume that our networks are connected (though not necessarily strongly connected) due to the fact that otherwise our network problems could be split into completely independent subproblems, with one subproblem per connected component of the network. Since the topic of this thesis are flows over time with commodity-dependent transit times we will focus almost exclusively on problems where multiple *commodities* are given. From now on, we will assume that we are given a set of commodities $K = \{1, \ldots, k\}$. Consequently, we denote the number of commodities by $k = |K|$.

Depending on the specific problem we are considering, we have several attributes on the nodes and arcs of our networks. We will introduce the various attributes here and specify later which attributes are used for which problem. *Capacities* $u_a \geq 0$ are attributes on every arc $a \in A$ limiting the amount of flow that can be sent through a. We refer to the entirety of capacities $u_a, a \in A$ as u for convenience. For every commodity $i \in K$, there can be *transit times* $\tau_a^i \geq 0$ for every arc $a \in A$ denoting the amount of time flow from commodity i needs to traverse through a. Once again, we refer to the entirety of transit times for a commodity $i \in K$ as τ^i and to the entirety of all transit times as τ. For a commodity $i \in K$, *sources* $S_+^i \subseteq V$ and *sinks* $S_-^i \subseteq V$ are sets of nodes with $S_+^i \cap S_-^i = \emptyset$ where flow originates and terminates, respectively. For convenience, we define $S^i := S_+^i \cup S_-^i$. *Supplies* and *demands* $b_v^i, i \in K, v \in V$ assign a supply (if $b_v^i > 0$) or a demand (if $b_v^i < 0$) to nodes $v \in V$ for every commodity $i \in K$. b^i and b denote the entirety of supplies and demands for a commodity $i \in K$ and all commodities, respectively. We assume that $b_v^i = 0$ for all $v \in V \setminus S^i$ and that the following equation holds for all commodities $i \in K$:

$$\sum_{v \in V} b_v^i = 0$$

When using supplies and demands we define sets of sources S_+^i and sinks S_-^i for every commodity i by

$$S_+^i := \left\{ v \in V \mid b_v^i > 0 \right\}$$
$$S_-^i := \left\{ v \in V \mid b_v^i < 0 \right\}$$

For all nodes $v \in V$, we define the following notations:

$$\delta^+(v) := \{ e = (v, w) \in A \mid w \in V \}$$
$$\delta^-(v) := \{ e = (w, v) \in A \mid w \in V \}$$
$$\delta(v) := \delta^+(v) \cup \delta^-(v)$$

Moreover, we extend these notations to all subsets $V' \subseteq V$:

$$\delta^+(V') := \left\{ e = (v, w) \in A \mid v \in V', w \in V \setminus V' \right\}$$
$$\delta^-(V') := \left\{ e = (w, v) \in A \mid v \in V', w \in V \setminus V' \right\}$$
$$\delta(V') := \delta^+(V') \cup \delta^-(V')$$

For a set of nodes $W \subseteq V$ the set of edges $\delta(W) \subseteq A$ is called the *(undirected) cut* induced by W. Likewise, $\delta^+(W)$ is called the *directed cut* induced by W. A directed cut $\delta^+(W)$ with nodes $s, t \in V$ is called *s-t-cut* if $s \in W$ and $t \notin W$. Furthermore, in settings with capacities $u_a, a \in A$ and transit times $\tau_a^i, a \in A, i \in K$, respectively, we define for all subsets $A' \subseteq A$ and all commodities $i \in K$:

$$u_{min}(A') := \min_{a \in A'} \{u_a\}$$
$$u(A') := \sum_{a \in A'} u_a$$
$$\tau^i(A') := \sum_{a \in A'} \tau_a^i$$

In settings with supplies and demands $b_v^i, v \in V, i \in K$ we define for all subsets $W \subseteq V$:

$$b^i(W) := \sum_{v \in W} b_v^i$$

Furthermore, we define

$$B^i := \sum_{v \in V: b_v^i > 0} b_v^i$$
$$B := \sum_{i \in K} B^i$$

We will now use these definitions and notations to define flows and transshipments. We begin with defining static flows and transshipments which have no concept of transit times. As before, we assume that we are given a network $G = (V, A)$ and a set of commodities $K = \{1, \ldots, k\}$ with capacities $u_a, a \in A$ and either sets of sources and sinks $S_+^i, S_-^i \subseteq V$ or supplies and demands $b_v^i, v \in V$ for every commodity $i \in K$.

Definition 2.1.2 (Flows)**.** *A* (static) flow *x in a network $G = (V, A)$ with sources $S_+ \subseteq V$ and sinks $S_- \subseteq V$ assigns a flow value $x_a \geq 0$ to every arc $a \in A$ such that* flow conservation *is fulfilled:*

$$\sum_{a \in \delta^-(v)} x_a = \sum_{a \in \delta^+(v)} x_a \quad \forall v \in V \setminus (S_+ \cup S_-)$$
$$\sum_{a \in \delta^-(v)} x_a \leq \sum_{a \in \delta^+(v)} x_a \quad \forall v \in S_+$$
$$\sum_{a \in \delta^-(v)} x_a \geq \sum_{a \in \delta^+(v)} x_a \quad \forall v \in S_-$$

*In the special case of $S_+ = \{s\}, S_- = \{t\}$ we call x an s-t-*flow*. A* (static) multicommodity flow *in a network $G = (V, A)$ with sources $S_+^i \subseteq V$ and sinks $S_-^i \subseteq V$ assigns a flow value $x_a^i \geq 0$ to every arc $a \in A$ for every commodity $i \in K$ such that* flow conservation *is fulfilled for all commodities $i \in K$:*

$$\sum_{a \in \delta^-(v)} x_a^i = \sum_{a \in \delta^+(v)} x_a^i \quad \forall v \in V \setminus (S_+^i \cup S_-^i)$$
$$\sum_{a \in \delta^-(v)} x_a^i \leq \sum_{a \in \delta^+(v)} x_a^i \quad \forall v \in S_+^i$$
$$\sum_{a \in \delta^-(v)} x_a^i \geq \sum_{a \in \delta^+(v)} x_a^i \quad \forall v \in S_-^i$$

Note that for every commodity $i \in K$ the flow values $x_a^i, a \in A$ define a (single-commodity) flow.

Definition 2.1.3 (Transshipments)**.** *A* (static) transshipment *x in a network $G = (V, A)$ with supplies and demands $b_v, v \in V$ assigns a flow value $x_a \geq 0$ to every arc $a \in A$ such that* flow conservation *is fulfilled:*

$$\begin{aligned} \sum_{a \in \delta^-(v)} x_a &= \sum_{a \in \delta^+(v)} x_a & \forall v \in V : b_v = 0 \\ 0 \leq \sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a &\leq b_v & \forall v \in V : b_v > 0 \\ 0 \leq \sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a &\leq -b_v & \forall v \in V : b_v < 0 \end{aligned}$$

A (static) multicommodity transshipment *in a network $G = (V, A)$ with supplies and demands $b_v^i, v \in V, i \in K$ assigns a flow value $x_a^i \geq 0$ to every arc $a \in A$ for every commodity $i \in K$ such that* flow conservation *is fulfilled for all commodities $i \in K$:*

$$\begin{aligned} \sum_{a \in \delta^-(v)} x_a^i &= \sum_{a \in \delta^+(v)} x_a^i & \forall v \in V : b_v^i = 0 \\ 0 \leq \sum_{a \in \delta^+(v)} x_a^i - \sum_{a \in \delta^-(v)} x_a^i &\leq b_v^i & \forall v \in V : b_v^i > 0 \\ 0 \geq \sum_{a \in \delta^+(v)} x_a^i - \sum_{a \in \delta^-(v)} x_a^i &\geq b_v^i & \forall v \in V : b_v^i < 0 \end{aligned}$$

Note that for every commodity $i \in K$ the flow values $x_a^i, a \in A$ define a (single-commodity) transshipment.

We will now introduce some concepts for single- and multicommodity flows and transshipments. Since single-commodity flows and transshipments are the special case $K = \{1\}$ of multicommodity transshipments we just define them for the case of multiple commodities.

A multicommodity flow or transshipment x in a network $G = (V, A)$ is called *feasible* in a setting with capacities $u_a, a \in A$ on the arcs if and only if

$$\sum_{i \in K} x_a^i \leq u_a \quad \forall a \in A$$

In a setting with supplies and demands, a multicommodity transshipment x *satisfies supplies and demands* b, if and only if

$$\sum_{a \in \delta+(v)} x_a^i - \sum_{a \in \delta+(v)} x_a^i = b_v^i \quad \forall v \in V, i \in K$$

The *value* $|x^i|$ of a commodity $i \in K$ in a multicommodity flow x is defined as

$$|x^i| := \sum_{a \in \delta^+(S_+^i)} x_a^i - \sum_{a \in \delta^-(S_+^i)} x_a^i$$

Likewise the *value* $|x|$ of a multicommodity flow x is defined as

$$|x| := \sum_{i \in K} |x^i|$$

By incorporating transit times into these definitions we can define flows and transshipments over time. In addition to the attributes listed above we are now also given transit times $\tau_a \geq 0$ for every arc $a \in A$ and a *time horizon* $T \geq 0$. In the following definitions, we will consider flows and transshipments over time which assign *flow rates* $f_a(t)$ to an arc $a \in A$ at a point in time t. In order to simplify notations, we will assume that $t \in \mathbb{R}$. However, we demand that $f_a(t) = 0$ for all $t \notin [0, T - \tau_a)$. We say that $f_a(t)$ is the rate at which flow enters the tail of arc a at time t. Flow entering the tail of a at time t will leave the head of a at time $t + \tau_a$.

Definition 2.1.4 (Flows over Time)**.** *A* flow over time f *in a network* $G = (V, A)$ *with sources* $S_+ \subseteq V$ *and sinks* $S_- \subseteq V$ *assigns a flow value* $f_a(t) \geq 0$ *to every arc* $a \in A$ *at every point in time* $t \in \mathbb{R}$ *such that* flow conservation *is fulfilled:*

$$\sum_{a \in \delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a(\theta)\ d\theta \geq \sum_{a \in \delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \quad \forall v \in V \setminus S_+, t \in [0, T]$$

$$\sum_{a \in \delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a(\theta)\ d\theta \leq \sum_{a \in \delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \quad \forall v \in S_+, t \in [0, T]$$

$$\sum_{a \in \delta^-(v)} \int_{\theta=0}^{T-\tau_a} f_a(\theta)\ d\theta = \sum_{a \in \delta^+(v)} \int_{\theta=0}^{T} f_a(\theta)\ d\theta \quad \forall v \in V \setminus (S_+ \cup S_-)$$

Note that we allow the storage *of flow at intermediate nodes as long as flow leaves the node before the time horizon is over. In the special case of* $S_+ = \{s\}, S_- = \{t\}$ *we call* f *an* s-t-flow over time*. A* multicommodity flow over time *in a network* $G = (V, A)$ *with sources* $S_+^i \subseteq V$ *and sinks* $S_-^i \subseteq V$ *assigns a flow value* $f_a^i(t) \geq 0$ *to every arc* $a \in A$ *for every commodity* $i \in K$ *at every point in time* $t \in \mathbb{R}$ *such that* flow conservation *is fulfilled for all commodities* $i \in K$*:*

$$\sum_{a \in \delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a^i(\theta)\ d\theta \geq \sum_{a \in \delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \quad \forall v \in V \setminus S, t \in [0, T]$$

$$\sum_{a \in \delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a^i(\theta)\ d\theta \leq \sum_{a \in \delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \quad \forall v \in S, t \in [0, T]$$

$$\sum_{a \in \delta^-(v)} \int_{\theta=0}^{T-\tau_a} f_a^i(\theta)\ d\theta = \sum_{a \in \delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\ d\theta \quad \forall v \in V \setminus (S \cup T)$$

Note that for every commodity $i \in K$ *the flow values* $f_a^i(t), a \in A, t \in \mathbb{R}$ *define a (single-commodity) flow over time.*

Definition 2.1.5 (Transshipments over Time)**.** *A* transshipment over time f *in a network* $G = (V, A)$ *with supplies and demands* $b_v, v \in V$ *assigns a flow value* $x_a \geq 0$ *to every arc* $a \in A$ *at every point in time* $t \in \mathbb{R}$ *such that* flow conservation *is fulfilled:*

$$\begin{aligned}
\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a(\theta)\ d\theta &\geq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta && \forall v \in V : b_v = 0, t \in [0, T] \\
\sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a} f_a(\theta)\ d\theta &= \sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a(\theta)\ d\theta && \forall v \in V : b_v = 0 \\
0 \leq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t-\tau_a} f_a(\theta)\ d\theta &- \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \leq b_v && \forall v \in V : b_v > 0, t \in [0, T] \\
0 \leq \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a(\theta)\ d\theta &- \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \leq -b_v && \forall v \in V : b_v < 0, t \in [0, T]
\end{aligned}$$

Once again we allow the storage *of flow at intermediate nodes as long as flow leaves before the time horizon is over. A* multicommodity transshipment over time *in a network* $G = (V, A)$ *with supplies and demands* $b_v^i, v \in V, i \in K$ *assigns a flow value* $f_a^i(t) \geq 0$ *to every arc* $a \in A$ *for every commodity* $i \in K$ *at every point in time* $t \in \mathbb{R}$ *such that* flow conservation *is fulfilled for all commodities* $i \in K$*:*

$$\begin{aligned}
\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a^i(\theta)\ d\theta &\geq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta && \forall v \in V : b_v^i = 0, t \in [0, T] \\
\sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a} f_a^i(\theta)\ d\theta &= \sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\ d\theta && \forall v \in V : b_v^i = 0 \\
0 \leq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t-\tau_a} f_a^i(\theta)\ d\theta &- \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \leq b_v^i && \forall v \in V : b_v^i > 0, t \in [0, T] \\
0 \leq \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a} f_a^i(\theta)\ d\theta &- \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \leq -b_v^i && \forall v \in V : b_v^i < 0, t \in [0, T]
\end{aligned}$$

Note that for every commodity $i \in K$ *the flow values* $f_a^i(t), a \in A, t \in \mathbb{R}$ *define a (single-commodity) transshipment over time.*

A multicommodity flow or transshipment over time f in a network $G = (V, A)$ is called *feasible* in a setting with capacities $u_a, a \in A$ on the arcs if and only if

$$\sum_{i\in K} f_a(t) \leq u_a \quad \forall a \in A, t \in [0, T]$$

In a setting with supplies and demands $b_v^i, v \in V, i \in K$, a multicommodity transshipment over time f *satisfies supplies and demands* b, if and only if

$$\sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\ d\theta - \sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a} f_a^i(\theta)\ d\theta = b_v^i \qquad \forall v \in V, i \in K$$

The *value* $|f^i(t)|$ of a commodity $i \in K$ in a multicommodity flow over time f up to a time t is defined as

$$|f^i(t)| := \sum_{a \in \delta^+(S_+^i)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta - \sum_{a \in \delta^-(S_+^i)} \int_{\theta=0}^{t-\tau_a} f_a^i(\theta)\ d\theta$$

Likewise the *value* $|f(t)|$ of a multicommodity flow over time f up to a time t is defined as

$$|f(t)| := \sum_{i \in K} |f^i|$$

For multicommodity transshipments over time f we define the $S_+^i := \left\{ v \in V \mid b_v^i > 0 \right\}$ and $S_-^i := \left\{ v \in V \mid b_v^i < 0 \right\}$ and apply the above definitions for *value* up to a time t. In a setting with a time horizon T we write $|f^i| := |f^i(T)|$ and $|f| := |f(T)|$.

Remark 2.1.6 (Loops). *As we can see by now, flow along loops $a = (v, v)$ has no effect on the value of a flow, on flow conservation or the fulfillment of supplies and demands. Therefore we can assume that our networks do not contain loops.*

We continue by introducing a very useful concept for computing flows - the residual network.

Definition 2.1.7 (Residual Network). *For a network $G = (V, A)$ with capacities u and a static flow or transshipment x in G we define for every arc $a = (v, w) \in A$ a* backward arc *$\overleftarrow{a} = (w, v) \notin A$ and $\overleftarrow{A} := \{\overleftarrow{a} \mid a \in A\}$. The backward arc of an arc $\overleftarrow{a} \in \overleftarrow{A}$ is $a \in A$. A* residual capacity *$u_{x,a}$ is assigned to every arc $a \in A \dot\cup \overleftarrow{A}$ given by*

$$u_{x,a} := \begin{cases} u_a - x_a & a \in A \\ x_a & a \in \overleftarrow{A} \end{cases}$$

The residual network *$G_x = (V_x, A_x)$ is defined by*

$$\begin{aligned} V_x &:= V \\ A_x &:= \left\{ a \in A \dot\cup \overleftarrow{A} \;\middle|\; u_{x,a} > 0 \right\} \end{aligned}$$

In cases where costs c_a or transit times τ_a are associated with arcs $a \in A$ we extend them to the backward arcs by defining $c_{\overleftarrow{a}} := -c_a$ and $\tau_{\overleftarrow{a}} := -\tau_a$.

In order to introduce the next concept we need to restrict our notion of time. More specifically, we need to abandon the concept of continuous time in favor of discrete time steps. This means that the flow values $f_a^i(t), a \in A, i \in K$ can only change at certain points in time t, e.g.

$$t \in \left\{ 0, \frac{1}{k}T, \frac{2}{k}T, \ldots, T \right\}, T \geq 0$$

We can interprete f_a^i as a stepwise constant function with $k+1$ steps in this case. Note that we do still assume that $f_a^i(t) = 0 \forall a \in A, i \in K$ for $t < 0$ or $t \geq T$. For $k \to \infty$, this model approaches the continuous time model. We will see that a discrete time model allows for some simplifications, while on the other hand - as shown by Fleischer and Tardos [9] - being close enough to the continuous time model that results in the discrete time model can be transferred to to the continuous time model, if this is deemed desireable.

Definition 2.1.8 (Time-Expanded Network). *For a network* $G = (V = \{v_1, \dots, v_n\}, A = \{a_1, \dots, a_m\}$ *with transit times* $\tau_a, a \in A$ *and a time steps* $0, 1, 2, \dots, T$, $T \in \mathbb{N}$ *we define the* time-expanded network $G^T = (V^T, A^T)$ *by*

$$\begin{aligned} V^T &:= \{ v_t \mid v \in V, t \in \{0, 1, \dots, T\}\} \\ A^T &:= A_1 \cup A_2 \\ A_1 &:= \{ a_t = (v_t, w_{t+\tau_a}) \mid a = (v, w) \in A, t \in \{0, \dots, T - \tau_a\}\} \\ A_2 &:= \{ a_{v,t} = (v_t, v_{t+1}) \mid v \in V, t \in \{0, \dots, T - 1\}\} \end{aligned}$$

Thus, we create T *copies of a node* $v \in V$ *and* $T - \tau_a$ *copies of an arc* $a \in A$*. Furthermore, we create* $T - 1$ holdover arcs *between copies of a node. Capacities or costs on arcs of the original network can easily be extended to the time-expanded network; to each copy of an arc the same cost and the same capacity are assigned as to the arc it derives from in the original network and holdover arcs are given unlimited capacities and no costs. Let* u' *and* c' *be the extended capacities and costs, respectively:*

$$u'_a := \begin{cases} u_a & a = a_t \in A_1 \\ \infty & a \in A_2 \end{cases}$$

$$c'_a := \begin{cases} c_a & a = a_t \in A_1 \\ 0 & a \in A_2 \end{cases}$$

Supplies and demands $b^i_v, v \in V, i \in K$ *are extended as follows:*

$$b^i_{v_t} := \begin{cases} b^i_v & b^i_v > 0, t = 0 \\ b^i_v & b^i_v < 0, t = T \\ 0 & else \end{cases} \quad \forall v_t \in V^T, i \in K$$

Note that every static flow or transshipment x in a time-expanded network $G^T = (V^T, A^T = A_1 \cup A_2)$ as defined above corresponds to a multicommodity flow or transshipment over time f with time horizon T in $G = (V, A)$ when using a discrete time model with time steps $0, 1, \dots, T$. In order to see this, we define for a given multicommodity flow or transshipment x in $G^T = (V^T, A^T = A_1 \cup A_2)$:

$$f^i_a(t) := \begin{cases} 0 & t \notin \{0, 1, \dots, T - \tau_a - 1\} \\ x^i_{a_t} & t \in \{0, 1, \dots, T - \tau_a - 1\} \end{cases} \forall i \in K, a \in A, t \in \{0, 1, \dots, T\}$$

On the other hand, based on a multicommodity flow over transshipment over time f within the discrete time model specified above, we define:

$$x_a^i := \begin{cases} f_a^i(t) & a = a_t \in A_1 \\ \sum_{a\in\delta^-(v)} f_a^i(t-\tau_a) - \sum_{a\in\delta^+(v)} f_a^i(t) & a = (v_t, v_{t+1}) \in A_2 \end{cases} \quad \forall i \in K, a \in A^T$$

It is easy to check that the static multicommodity flow or transshipment x in G^T is equivalent to the multicommodity flow or transshipment over time f in G in the sense that there is the bijection given above which preserves the value, feasibility, fulfillment of supplies and demands. Due to the fact that our focus lies elsewhere we leave this to the reader.

We conclude this section with this definition and continue by introducing common problems related to network flows.

2.2 Common Problems

In this section we will define classic network flow problems that we want to extend to commodity-dependent transit times. We will define the problems by specifying the input the problem is based on as well as the task the problem consists of. For a given network flow problem P, we call a specific input I for this problem an *instance* of P. The *size* $size(I)$ of an instance I usually depends on the number of nodes and edges of the underlying network, the number of commodities and the size of numbers appearing as attributes. For problems with a temporal dimension, the size also depends on the time horizon.

We begin with one of the most basic flow problems, the Maximum Flow Problem proposed by Ford & Fulkerson [10]. This problem does not incorporate a temporal dimension, it just consists of sending as much flow as possible from sources to sinks in a static network. There is only a single commodity.

Maximum Flow Problem	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$; a set of sources $S \subseteq V$ and a set of sinks $T \subseteq V$ with $S \cap T = \emptyset$.
Task:	Find a feasible flow x of maximum value $\|x\|$.

Without loss of generality, we can assume that $S = \{s\}$ und $T = \{t\}$. The size of an instance I of the Maximum Flow Problem is linear in $|V|$, $|A|$ and $\log u(A)$.

Lemma 2.2.1 (Supersource and Supersink). *For every instance $I = (G = (V, A), u, S, T)$ with $|S| > 1$ or $|T| > 1$ there is an instance $I' = (G' = (V', A'), u', S' = \{s\}, T' = \{t\})$ such that an optimal solution x' of I' can be transformed into an optimal solution x of I. Furthermore, the size of I' is polynomial in size(I).*

Proof. Consider an instance $I = (G = (V, A), u, S, T)$ of the maximum flow problem not fulfilling this property, i.e. $S = \{s_1, \dots, s_i\}$ and $T = \{t_1, \dots, t_j\}$ with either $i >$ or $j > 1$.

We create an instance $I' = (G' = (V', A'), u', S', T')$ by adding a *supersource* s and a *supersink* t which are connected to the sources and sinks, respectively:

$$V' := V \cup \{s, t\}$$
$$A' := A \cup \{\,(s, v) \mid v \in S\} \cup \{\,(v, t) \mid v \in T\}$$
$$u'_a := \begin{cases} u_a & a \in A \\ u(\delta^+(v)) & a = (s, v) \in A' \\ u(\delta^-(v)) & a = (v, t) \in A' \end{cases}$$
$$S' := \{s\}$$
$$T' := \{t\}$$

The new instance I' has $|V| + 2$ nodes, at most $|A| + |V|$ edges and all capacities u'_a are bounded by $u(A)$. For any solution x of I we can define a solution x' of I' with $|x| = |x'|$ by:

$$x'_a := \begin{cases} x_a & a \in A \\ x(\delta^+(v)) & a = (s, v) \in A' \\ x(\delta^-(v)) & a = (v, t) \in A' \end{cases}$$

x' is feasible flow for I' with $|x| = |x'|$ by construction. On the other hand, we can define a solution x of I based on a solution x' of I' with $|x| = |x'|$ by:

$$x_a := x'_a \qquad a \in A$$

Once again, x is feasible flow for I with $|x| = |x'|$ by construction. Thus, we have a flow-value preserving bijection between the feasible flows of I and I'. Therefore, it must map optimal solutions of I to optimal solution of I' and vice versa. □

We can also employ this construction to ensure that $\delta^-(s) = \delta^+(t) = \emptyset$. Thus we can assume without loss of generality that s has no incoming arcs and t no outgoing arcs. A straightforward way to solve an instance $I = (G = (V, A), u, S = \{s\}, T = \{t\})$ of the Maximum Flow Problem consists of formulating it as a LP. Assuming $\delta^-(s) = \delta^+(t) = \emptyset$, this can be done as follows:

$$\begin{array}{lll} \max & |x| = x(\delta^+(s)) = x(\delta^-(t)) & \\ \text{s.t.} & x(\delta^-(v)) = x(\delta^+(v)) & v \in V \setminus \{s, t\} \\ & x_a \le u_a & a \in A \\ & x_a \ge 0 & a \in A \end{array}$$

In practice, there are more efficient combinatorial strategies, as we will see in the next chapter.

We conclude our consideration of the Maximum Flow Problem for now and introduce the related Transshipment Problem. It differs from the Maximum Flow

PROBLEM only in the fact that we are given specific supplies and demands for the sources and sinks and need to decide whether a feasible flow fulfilling all supplies and demands exists.

TRANSSHIPMENT PROBLEM	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$ and supplies and demands b_v for every node $v \in V$.
Task:	Find a feasible transshipment x fulfilling the supplies and demands b.

The size of a transshipment problem is linear in $|V|$, $|A|$, $\log u(A)$ and $\log B$. This problem is closely related to the MAXIMUM FLOW PROBLEM, as we will see in the next lemma.

Lemma 2.2.2. *An instance $I = (G = (V, A), u, b)$ of the* TRANSSHIPMENT PROBLEM *can be solved by reducing it to an instance $I' = (G' = (V', A'), u', S, T)$ of the* MAXIMUM FLOW PROBLEM *with size(I') polynomial in size(I).*

Proof. We construct I' by defining

$$V' := V \cup \{s, t\}$$
$$A' := A \cup \{\, (s, v) \mid v \in S\} \cup \{\, (v, t) \mid v \in T\}$$
$$u'_a := \begin{cases} u_a & a \in A \\ b_v & a = (s, v) \in A' \\ -b_v & a = (v, t) \in A' \end{cases}$$
$$S' := \{s\}$$
$$T' := \{t\}$$

If a solution x' of I' with maximum value $|x'| = \sum_{v \in V : b(v) > 0} b(v)$ exists, then we can define a feasible flow x for I that fulfills all supplies and demands by restricting x' to A. This works due to the construction of the network - $|x'| = \sum_{v \in V : b(v) > 0} b(v)$ implies that $x'_a = u'_a = b_{head(a)}$ for all $a \in A' : tail(a) = s$ and $x'_a = u'_a = -b_{tail(a)}$ for all $a \in A' : head(a) = t$. Therefore x is a feasible flow that fulfills all supplies and demands. □

If the integrality of the optimal solution's value is guaranteed for an instance I of the MAXIMUM FLOW PROBLEM we can also easily solve I by reducing it to the TRANSSHIPMENT PROBLEM. We add a supply b to s and a demand $-b$ to t and thereby create an instance I_b of the TRANSSHIPMENT PROBLEM. For $b = 0, \ldots, u(S)$ we obtain a family of instances $\mathcal{I} = \{I_b \mid b = 0, \ldots, u(S)\}$ of the TRANSSHIPMENT PROBLEM and perform binary search on the greatest value b for which an instance I_b has a feasible solution. This can be done in $\log u(S)$ time which is still polynomial in $size(I)$.

Similar to the MAXIMUM FLOW PROBLEM we can obtain an LP formulation for the TRANSSHIPMENT PROBLEM:

$$
\begin{array}{lll}
\max & 1 & \\
\text{s.t.} & x(\delta^+(v)) - x(\delta^-(v)) = 0 & v \in V \setminus \{s,t\} \\
& x_a \leq u_a & a \in A \\
& x_a \geq 0 & a \in A
\end{array}
$$

We can actually use any objective function, since we are only interested in whether a feasible solution to the LP exists. It is usually more efficient to employ the reduction to the MAXIMUM FLOW PROBLEM, however.

Adding costs c_a to the arcs $a \in A$ leads to another classic network flow problem, the MINIMUM COST FLOW PROBLEM.

MINIMUM COST FLOW PROBLEM

Instance: A network $G = (V, A)$ with capacities u_a and costs c_a for every arc $a \in A$; supplies and demands b_v for every node $v \in V$.

Task: Find a feasible flow x fulfilling the supplies and demands b that minimizes $\sum_{a \in A} c_a x_a$.

We conclude our definitions of network flow problems without transit times here and move on to network flow problems with a temporal dimension. We still consider the special case of one commodity for now, however. Let us begin by extending the MAXIMUM FLOW PROBLEM to transit times:

MAXIMUM FLOW OVER TIME PROBLEM

Instance: A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$; a set of sources $S_+ \subseteq V$, a set of sinks $S_- \subseteq V$ with $S_+ \cap S_- = \emptyset$ and a time horizon $T \geq 0$.

Task: Find a feasible flow over time f with time horizon T of maximum value $|f|$.

Similar to the static case, we can introduce a supersource s and supersink t to ensure that $S_+ = \{s\}$ and $S_- = \{t\}$ with $\delta^-(s) = \emptyset$ and $\delta^+(t) = \emptyset$ holds. We cannot use a linear program to solve this problem in a continuous time model, though. But if we use a discrete time model we can construct a time-expanded network; due to the bijection between flows over time in the original network and static flows in the time-expanded network we can solve a MAXIMUM FLOW PROBLEM in the time-expanded network to solve the MAXIMUM FLOW OVER TIME PROBLEM in the original network. The size of the time-expanded network is linear in T, however - this method has therefore a pseudopolynomial runtime.

We will see in the next chapter that more sophisticated polynomial time approaches are possible.

We continue by doing the same for the Transshipment Problem, yielding the Transshipment over Time Problem:

Transshipment over Time Problem	
Instance:	A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$ and supplies and demands b_v for every node $v \in V$; a time horizon $T \geq 0$.
Task:	Find a feasible transshipment over time f with time horizon T fulfilling all supplies and demands b.

Contrary to the static Transshipment Problem it is not easily possible to reduce the Transshipment over Time Problem to the Maximum Flow over Time Problem. This is due to the fact that capacities in the static case bound the *total amount* of flow passing through an arc. For flows and transshipments over time, a capacity u_a for an arc a limits only the rate at which flow enters a at a time t but not directly the total amount of flow passing through a. Therefore it is not possible to use the same reduction as in the static case. When using a discrete time model we can reduce the Transshipment over Time Problem to a Transshipment Problem in a similar manner as for the Maximum Flow over Time Problem and the Maximum Flow Problem, respectively. We will see more interesting approaches in the next chapter.

The Quickest Transshipment Problem is an optimization problem related to the Transshipment over Time Problem. Whereas the Transshipment over Time Problem is just interested in whether a feasible solution exists, the Quickest Transshipment Problem asks for the minimum time horizon $T \geq 0$ such that a feasible solution exists:

Quickest Transshipment Problem	
Instance:	A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$ and supplies and demands b_v for every node $v \in V$.
Task:	Find the minimum time horizon $T \geq 0$ such that a feasible transshipment over time f with time horizon T fulfilling all supplies and demands b exists.

When using a discrete time model, this problem can be solved by solving instances of Transshipment over Time Problem for specific values of T. In conjunction with geometric or binary search, we are bound to find the optimal time horizon (due to time being discrete). Once again, we will see more approaches in the next chapter.

The next problem is motivated by evacuation optimization. In evacuation optimization it is not only desireable to evacuate all people as fast as possible but also to evacuate as many people as possible at every point in time, too - after all, we usually do not know how much time is available before a catastrophy happens.

In order to define the next problem, we need a concept of partially fulfilled supplies and demands. For this reason, we will define the value $|f|$ of a multicommodity transshipment over time f as

$$|f| := \sum_{i \in K} \sum_{v \in V : b_v^i > 0} \left(\sum_{a \in \delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\, d\theta - \sum_{a \in \delta^-(v)} \int_{\theta=0}^{T-\tau_a} f_a^i(\theta)\, d\theta \right)$$

This allows us to define the Earliest Arrival Transshipment Problem:

Earliest Arrival Transshipment Problem	
Instance:	A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$ and supplies and demands b_v for every node $v \in V$
Task:	Find the minimum time horizon $T \geq 0$ and a feasible transshipment over time f with time horizon T such that f fulfills all supplies and demands b and $\|f(t)\|$ is maximal at every point in time $t \in [0, T]$.

A transshipment with this property is called *earliest arrival transshipment.* Note that an earliest arrival transshipment does not always exist. In order to work with multicommodity flow problems, we extend these definitions to multiple commodities. We begin with the three static network flow problems.

Maximum Multicommodity Flow Problem	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$, a set of commodities $K = \{1, \dots, k\}$ and sets of sources $S_+^i \subseteq V$ and sets of sinks $S_-^i \subseteq V$ with $S_+^i \cap S_-^i = \emptyset$ for every commodity $i \in K$.
Task:	Find a feasible multicommodity flow x of maximum value $\|x\|$.

Multicommodity Transshipment Problem	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$, a set of commodities $K = \{1, \dots, k\}$ and supplies and demands b_v^i for every node $v \in V$ and every commodity $i \in K$.
Task:	Find a feasible multicommodity transshipment x fulfilling the supplies and demands b.

Minimum Cost Multicommodity Flow Problem

Instance: A network $G = (V, A)$ with capacities u_a and costs c_a for every arc $a \in A$, a set of commodities $K = \{1, \dots, k\}$ and supplies and demands b_v^i for every node $v \in V$ and every commodity $i \in K$.

Task: Find a feasible multicommodity flow x fulfilling the supplies and demands b that minimizes $\sum_{a \in A} c_a \left(\sum_{i \in K} x_a^i \right)$.

The single-commodity problems are obviously special cases of these problems, therefore we cannot expect these problems to be solved more easily than their single commodity counterparts. In fact we will see in the next chapter that there are no exact polynomial time combinatorial algorithms known for them. We can still formulate them as linear programs, though. More on these problems in the next chapter.

We can extend the problems related to flows and transshipments over time to multiple commodities as well. Once again, the multiple commodity versions cannot be easier than the single commodity versions. In fact, we will see in the next chapter that the complexity of several problems increases when moving to the multicommodity case. In case of the Earliest Arrival Multicommodity Transshipment Problem we require the transshipment to fulfill as much supplies and demands as possible at every point in time. As we have already implied in the definition of a transshipment's value, we do not care how many supplies and demands from a specific commodity have been fulfilled but only about the total value.

Maximum Multicommodity Flow over Time Problem

Instance: A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$, a set of commodities $K = \{1, \dots, k\}$ and sets of sources $S_+^i \subseteq V$ and sets of sinks $S_-^i \subseteq V$ with $S_+^i \cap S_-^i = \emptyset$ for every commodity $i \in K$; a time horizon $T \geq 0$.

Task: Find a feasible multicommodity flow over time f with time horizon T of maximum value $|f|$.

Multicommodity Transshipment over Time Problem

Instance: A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$, a set of commodities $K = \{1, \dots, k\}$ and supplies and demands b_v^i for every node $v \in V$ and every commodity $i \in K$; a time horizon $T \geq 0$.

Task: Find a feasible multicommodity transshipment over time f with time horizon T fulfilling all supplies and demands b.

Quickest Multicommodity Transshipment Problem

Instance: A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$, a set of commodities $K = \{1, \ldots, k\}$ and supplies and demands b_v^i for every node $v \in V$ and every commodity $i \in K$.

Task: Find the minimum time horizon $T \geq 0$ such that a feasible multicommodity transshipment over time f with time horizon T fulfilling all supplies and demands b exists.

Earliest Arrival Multicommodity Transshipment Problem

Instance: A network $G = (V, A)$ with capacities u_a and transit times τ_a for every arc $a \in A$, a set of commodities $K = \{1, \ldots, k\}$ and supplies and demands b_v^i for every node $v \in V$.

Task: Find the minimum time horizon $T \geq 0$ and a feasible multicommodity transshipment over time f with time horizon T such that f fulfills all supplies and demands b and $|f(t)|$ is maximal at every point in time $t \in [0, T]$.

We conclude the definitions with these problems. In the next chapter we will take a look at already known solutions for these problems.

Chapter 3

Known Results

In this chapter we will discuss already known results to the problems defined in the previous chapter. We begin with the static single-commodity flow problems.

The Maximum Flow Problem was originally proposed by Ford and Fulkerson [11]. They were also able to provide an algorithm for this problem that relies on finding source-sinks-paths in the residual network. Their algorithm is guaranteed to find a solution if all capacities are integer; if irrational capacities are allowed it is possible that the algorithm does not terminate. Even in the case of integral capacities it is possible that the number of iterations (i.e. the number of paths required) is linear in the value of the maximum flow - thus, the algorithm has only pseudopolynomial runtime even in this case. Since then, many other improved algorithms have been proposed. A theoretically and practically very fast algorithm is due to Goldberg and Tarjan [14]. It uses a dual approach to the problem: it begins with a mapping of flow values to edges that violates flow conservation but has a value at least as great as a maximum flow. Afterwards, it stepwise modifies the mapping in a way that on the one hand guarantees that the value of the mapping remains at least as high as the value of a maximum flow and on the other hand improves the flow conservation of the mapping. This approach leads to a polynomial time algorithm, regardless of capacities.

The Transshipment Problem can be reduced to the Maximum Flow Problem very efficiently. Therefore any algorithm for the Maximum Flow Problem can solve the Transshipment Problem as well. The algorithm by Goldberg and Tarjan would usually be a good choice due to its efficiency.

The Minimum Cost Flow Problem can be attacked in various ways. Two of the most prominent approaches are cycle cancelling algorithms and successive shortest path algorithms, which employ primal and dual strategies, respectively, to tackle the problem. Cycle cancelling algorithms begin with a transshipment that fulfills all supplies and demands by solving a Transshipment Problem. Afterwards, the algorithms look for cycles of negative total cost in the residual network of the transshipment and augment flow along such cycles. This results in a new transshipment that still fulfills all supplies and demands but is of lower cost than the original one. Tardos [28] was able to show that for a suitable cycle selection strategy, the number of cycle augmentations is polynomial. A successive shortest path algorithm starts with a transshipment of at most optimal cost that usually does not fulfill all supplies and demands. Afterwards it is modified by augmenting flow along shortest paths in the residual network. In its basic form, this algorithm is of pseudopolynomial runtime (it depends on the total number of supplies). However, it is possible to guarantee a polynomial runtime by employing capacity scaling (see Edmonds and Karp [4] for details).

We move on to the problems related to single-commodity flows and transshipments over time.

The Maximum Flow over Time Problem was proposed and solved by Ford and Fulkerson [10, 11]. They were able to provide two different approaches to the problem. The first approach requires discrete time and consists of solving a Maximum Flow Problem in the time-expanded network. This strategy is only of pseudopolymial runtime but allows us to transfer all algorithms for the Maximum Flow Problem directly to the Maximum Flow over Time Problem. The second approach requires to introduce a supersource and a supersink as described in the previous chapter. Then it is possible to reduce the Maximum Flow over Time Problem to the Minimum Cost Flow Problem by interpreting the transit times as costs. An additional arc between supersink and supersouce of unlimited capacity and cost $-T$ is added (where T is the time horizon of the Maximum Flow over Time Problem) and a minimum cost circulation (i.e. a transshipment where supplies and demands are zero for every node) is computed. Afterwards, a maximum flow over time can be extracted. This is a polynomial time algorithm, since there are polynomial time algorithms for the Minimum Cost Flow Problem. The reduction can obviously be done in polynomial time as well.

Contrary to the static case, the Transshipment over Time Problem cannot be reduced to the Maximum Flow over Time Problem easily. Therefore, we can still employ a time-expanded network in the case of discrete time to reduce the problem to the Transshipment Problem. This constitutes a pseudopolynomial time algorithm for the problem. Hoppe and Tardos [20] were able to show that there are polynomial time algorithms for the Transshipment over Time Problem. Their algorithm requires an oracle for submodular function minimization, however. Submodular function minimization has been shown to be solvable in polynomial time using the ellipsoid method by Grötschel, Lovasz and Schrijver [16, 15]. This strategy is too slow to be usable in practice, unfortunately - the original algorithm has a worst case running time of $O(n^7 + Fn^5)$ where F is the time required to evaluate the submodular function. Although this running time has been improved upon by others, it is still prohibitively high.

Although we did not define the Minimum Cost Flow Problem for flows over time it is easy to realize how this would be done. The Minimum Cost Flow over Time

PROBLEM is known to be NP-hard, contrary to the MINIMUM COST FLOW PROBLEM. The proof of NP-hardness is due to Klinz and Woeginger [21] and consists of a reduction of the MINIMUM COST FLOW OVER TIME PROBLEM to the length bounded shortest path problem. In the case of discrete time, there are still pseudopolynomial time algorithms - we can create the time-expanded network and solve a MINIMUM COST FLOW PROBLEM on it. Fleischer and Skutella [7] gave a fully polynomial time approximation scheme for this problem that consists of condensing the time-expanded network to a size that on the one hand is large enough to obtain a solution of sufficient quality and on the other hand is small enough to be polynomial in the input size.

The QUICKEST TRANSSHIPMENT PROBLEM can be solved in pseudopolynomial time by employing time-expanded networks as well, as we already described in the previous chapter. Hoppe and Tardos [20] described a polynomial time algorithm which again requires an oracle for submodular function minimization. In case of only a single node with supplies and a single node with demands it is possible to solve instances of the MAXIMUM FLOW OVER TIME PROBLEM with varying time horizons and use a search strategy in order to find the optimal time horizon. Condensed time-expanded networks can be used to obtain a fully polynomial time approximation scheme for this problem as shown by Fleischer and Skutella [6].

The EARLIEST ARRIVAL TRANSSHIPMENT PROBLEM is the first problem where we cannot be certain that a solution exists. Gale [12] shows that an earliest arrival flow exists if there is only a single node with supply and a single node with demand. Richardson and Tardos [25] extended this proof of existence to mutiple nodes with supply and a single node with demands. For instances of the problem where more than a single node has demands there are simple counterexamples for the existence of earliest arrival transshipments. The following counterexample is due to Fleischer [5]. Consider the network $G = (V, A)$ defined by

$$V := \{s, v, t_1, t_2\}$$
$$A := \{e = (s, v), f = (v, t_1), g = (v, t_2)\}$$

We add capacities $u_e = u_f = u_g = 1$ and transit times $\tau_e = \tau_f = 0$ and $\tau_g = 1$. Furthermore we define supplies and demands by $b_s = 2, b_v = 0, b_{t_1} = b_{t_2} = -1$.

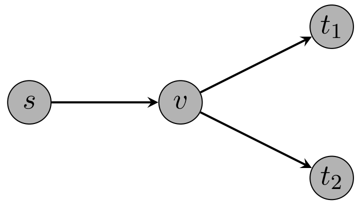


Figure 3.1: Counterexample to the existence of earliest arrival flows.

The structure of the network is shown in figure 3.1. There are two paths that can be used for routing flow from sources to sinks in this network, $P = s \to v \to t_1$ and $Q = s \to v \to t_2$. Obviously, $u_{min}(P) = u_{min}(Q) = 1$ and $\tau(P) = 0$, $\tau(Q) = 1$ holds. If we send flow along path P during the time interval $[0, 1]$ and along Q during $[1, 2]$ then flow will arrive at the sinks with a rate of one during $[0, 1]$ (via P) and during $[2, 3]$ (via Q). But if we reverse the order in which we use the paths, i.e. if we send flow along Q during $[0, 1]$ and along P during $[1, 2]$ the result will be flow arriving at a rate of two during $[1, 2]$. Therefore, any earliest arrival transshipment would have to sens at least one unit of flow to the sinks by time 1 and two units of flow by time 2. It is easy to see that both is not possible.

For the case of a single sink, there are pseudopolynomial time algorithms by Minieka [24] and Wilkinson [29] that employ successive shortest path based techniques. To the knowledge of the author it is unknown whether this problem is NP-hard. It is also an open question whether always a solution can be found which can be encoded in length polynomial in the input size. Baumann and Skutella [3] were able to develop an algorithm for the single sink case that is polynomial in the input and the output size. This algorithm requires an oracle for submodular function minimization, though. Furthermore, Hoppe and Tardos [19] and Fleischer and Skutella [6] give fully polynomial time approximation schemes which are based on capacity scaling and condensed time-expanded networks, respectively.

Problem	Complexity	
Maximum Flow Problem	P	
Transshipment Problem	P	
Minimum Cost Flow Problem	P	
Maximum Flow over Time Problem	P	
Transshipment over Time Problem	P^*	FPTAS
Minimum Cost Flow over Time Problem	NP-hard	FPTAS
Quickest Transshipment Problem	P^*	FPTAS
Earliest Arrival Transshipment Problem	Unknown	FPTAS

Table 3.1: Complexity of single-commodity network problems.

We make a pause to summarize the results regarding the complexity of our problems in table 3.1. P and NP refer to the complexity classes as usual, P^* denotes problems where the only polynomial time algorithms known require oracles for submodular function minimization. FPTAS refers to the existence of a fully polynomial time approximation scheme.

We continue with static multicommodity flows. As mentioned in the previous chapter, we can formulate the Maximum Multicommodity Flow Problem, the Multicommodity Transshipment Problem and the Minimum Cost Multicommodity Flow

Problem as linear programs. In conjunction with the ellipsoid method this guarantees us polynomial time algorithms for these problems. Unfortunately, the ellipsoid is usually very slow in practice rendering this approach rather inviable. Contrary to the single commodity of this problems, there are no exact combinatorial algorithms known for these problems. There are however various approaches for fast approximation algorithms, see for example Garg and Koenemann [13].

We conclude this chapter with results to problems related to multicommodity flows and transshipments over time. The Maximum Multicommodity Flow over Time Problem can be solved by formulating a linear program on the time-expanded network (assuming discrete time), yielding a pseudopolynomial time algorithm for this problem. Hall, Hippler and Skutella [17] showed that the Multicommodity Transshipment over Time Problem is NP-hard by reducing the well-known Partition Problem to the Multicommodity Transshipment over Time Problem. On the one hand a linear program based on the time-expanded network can be formulated to solve it in pseudopolynomial time. The Minimum Cost Multicommodity Flow over Time Problem is NP-hard as well, since the Minimum Cost Flow over Time Problem is already NP-hard.

The Quickest Multicommodity Transshipment Problem is NP-hard as well; Hall, Hippler and Skutella [17] reduce the Partition Problem to it. Fleischer and Skutella [6] give a fully polynomial time approximation scheme based on condensed time expanded networks for this problem. The Earliest Arrival Multicommodity Transshipment Problem is not guaranteed to have a solution for the case of multiple sinks, we can simply look at the single commodity case to realize this. If all commodities share the same sink, we can reduce the problem to the single-commodity case since we have essentially only a single commodity then.

Problem	Complexity
Maximum Flow Problem	P
Transshipment Problem	P
Minimum Cost Flow Problem	P
Transshipment over Time Problem	NP-hard
Minimum Cost Flow over Time Problem	NP-hard
Quickest Transshipment Problem	NP-hard
Earliest Arrival Transshipment Problem	NP-hard

Table 3.2: Complexity of multicommodity network problems.

We conclude this chapter with an overview of the complexity results for multicommodity flow and transshipments problems in table 4.1. For reasons of brevity we omit the "multicommodity" from the problems' names in this table.

Chapter 4

Models

In this chapter we will introduce models for multicommodity network flow and transshipment problems with commodity-dependent transit times. As before, we assume that we are given a network $G = (V, A)$ and a set of commodities $K = \{1, \dots, k\}$ with capacities $u_a, a \in A$ and either sets of sources and sinks $S^i_+, S^i_- \subseteq V$ or supplies and demands $b^i_v, v \in V$ for every commodity $i \in K$. In contrast to the previous chapters we will use commodity-dependent transit times $\tau^i_a \geq 0$ for all $a \in A$, $i \in K$ instead of normal transit times $\tau_a \geq 0$ $\forall a \in A$. We will now examine how commodity-dependent transit times affect our concepts of flows and transshipments and how they effect the problems we introduced in Chapter 2.

4.1 Commodity-Dependent Transit Times

The most straight-forward way to do this is to assign transit times τ^i_a to every arc $a \in A$ and every commodity $i \in K$ of a network $G = (V, A)$.

Definition 4.1.1 (Commodity-Dependent Transit Times)**.** *Let $G = (V, A)$ be a network and $K = \{1, \dots, k\}$ a set of commodities.* Commodity-dependent transit times τ *assign a non-negative value τ^i_a to every arc $a \in A$ and every commodity $i \in K$.*

We will begin by taking a closer look at the aspects affected by the introduction of commodity-dependent transit times. We interprete a transit time τ^i_a as the amount of time flow of commodity i needs to traverse arc a. Thus, flow of commodity i entering the tail of a at time t will leave the head of a at time $t + \tau^i_a$. Since flow of different commodities can move through an arc at different speeds, we need to incorporate the commodity-dependent transit times into our defintions of flows and transshipments over time.

Definition 4.1.2 (Flows over Time with Commodity-Dependent Transit Times)**.** *A* flow over time with commodity-dependent transit times f *in a network $G = (V, A)$ with sources*

$S_+ \subseteq V$ and sinks $S_- \subseteq V$ assigns a flow value $f_a(t) \geq 0$ to every arc $a \in A$ at every point in time $t \in \mathbb{R}$ such that flow conservation *is fulfilled:*

$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a(\theta)\ d\theta \geq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \quad \forall v \in V \setminus S_+, t \in [0,T]$$
$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a(\theta)\ d\theta \leq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \quad \forall v \in S_+, t \in [0,T]$$
$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a^i} f_a(\theta)\ d\theta = \sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a(\theta)\ d\theta \quad \forall v \in V \setminus (S_+ \cup S_-)$$

Note that we allow the storage *of flow at intermediate nodes as long as flow leaves the node before the time horizon is over. In the special case of $S_+ = \{s\}, S_- = \{t\}$ we call f an s-t-*flow over time with commodity-dependent transit times. *A* multicommodity flow over time with commodity-dependent transit times *in a network $G = (V, A)$ with sources $S_+^i \subseteq V$ and sinks $S_-^i \subseteq V$ assigns a flow value $f_a^i(t) \geq 0$ to every arc $a \in A$ for every commodity $i \in K$ at every point in time $t \in \mathbb{R}$ such that* flow conservation *is fulfilled for all commodities $i \in K$:*

$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a^i(\theta)\ d\theta \geq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \quad \forall v \in V \setminus S, t \in [0,T]$$
$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a^i(\theta)\ d\theta \leq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \quad \forall v \in S, t \in [0,T]$$
$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a^i} f_a^i(\theta)\ d\theta = \sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\ d\theta \quad \forall v \in V \setminus (S \cup T)$$

Note that for every commodity $i \in K$ the flow values $f_a^i(t), a \in A, t \in \mathbb{R}$ define a (single-commodity) flow over time with commodity-dependent transit times.

Similary, we modify our definitions of transshipments over time.

Definition 4.1.3 (Transshipments over Time with Commodity-Dependent Transit Times)**.** *A* transshipment over time with commodity-dependent transit times *f in a network $G =$*

(V, A) *with supplies and demands* $b_v, v \in V$ *assigns a flow value* $x_a \geq 0$ *to every arc* $a \in A$ *at every point in time* $t \in \mathbb{R}$ *such that* flow conservation *is fulfilled:*

$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a(\theta)\ d\theta \geq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \qquad \forall v \in V : b_v = 0, t \in [0,T]$$

$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a^i} f_a(\theta)\ d\theta = \sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a(\theta)\ d\theta \qquad \forall v \in V : b_v = 0$$

$$0 \leq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t-\tau_a^i} f_a(\theta)\ d\theta - \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \leq b_v \qquad \forall v \in V : b_v > 0, t \in [0,T]$$

$$0 \leq \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a(\theta)\ d\theta - \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a(\theta)\ d\theta \leq -b_v \qquad \forall v \in V : b_v < 0, t \in [0,T]$$

Once again we allow the storage *of flow at intermediate nodes as long as flow leaves before the time horizon is over. A* multicommodity transshipment over time with commodity-dependent transit times *in a network* $G = (V, A)$ *with supplies and demands* $b_v^i, v \in V, i \in K$ *assigns a flow value* $f_a^i(t) \geq 0$ *to every arc* $a \in A$ *for every commodity* $i \in K$ *at every point in time* $t \in \mathbb{R}$ *such that* flow conservation *is fulfilled for all commodities* $i \in K$*:*

$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a^i(\theta)\ d\theta \geq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \qquad \forall v \in V : b_v^i = 0, t \in [0,T]$$

$$\sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a^i} f_a^i(\theta)\ d\theta = \sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\ d\theta \qquad \forall v \in V : b_v^i = 0$$

$$0 \leq \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t-\tau_a^i} f_a^i(\theta)\ d\theta - \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \leq b_v^i \qquad \forall v \in V : b_v^i > 0, t \in [0,T]$$

$$0 \leq \sum_{a\in\delta^-(v)} \int_{\theta=0}^{t-\tau_a^i} f_a^i(\theta)\ d\theta - \sum_{a\in\delta^+(v)} \int_{\theta=0}^{t} f_a^i(\theta)\ d\theta \leq -b_v^i \qquad \forall v \in V : b_v^i < 0, t \in [0,T]$$

Note that for every commodity $i \in K$ *the flow values* $f_a^i(t), a \in A, t \in \mathbb{R}$ *define a (single-commodity) transshipment over time with commodity-dependent transit times.*

A multicommodity flow or transshipment over time with commodity-dependent transit times f in a network $G = (V, A)$ is called *feasible* in a setting with capacities $u_a, a \in A$ on the arcs if and only if

$$\sum_{i\in K} f_a(t) \leq u_a \quad \forall a \in A, t \in [0,T]$$

In a setting with supplies and demands $b_v^i, v \in V, i \in K$, a multicommodity transshipment over time with commodity-dependent transit times$f$ *satisfies supplies and demands* b, if and only if

$$\sum_{a\in\delta^+(v)} \int_{\theta=0}^{T} f_a^i(\theta)\ d\theta - \sum_{a\in\delta^-(v)} \int_{\theta=0}^{T-\tau_a^i} f_a^i(\theta)\ d\theta = b_v^i \qquad \forall v \in V, i \in K$$

The *value* $|f^i|$ of a commodity $i \in K$ in a multicommodity flow over time with commodity-dependent transit times f is defined as

$$|f^i| := \sum_{a \in \delta^+(S^i_+)} \int_{\theta=0}^{T} f^i_a(\theta)\ d\theta - \sum_{a \in \delta^-(S^i_+)} \int_{\theta=0}^{T-\tau^i_a} f^i_a(\theta)\ d\theta$$

Likewise the *value* $|f|$ of a multicommodity flow over time with commodity-dependent transit times f is defined as

$$|f| := \sum_{i \in K} |f^i|$$

As we can see, the capacity constraints remain unchanged by the introduction of commodity-dependent transit times. The flow conservation, supply-demands constraints and the definition of the value of a commodity are slightly modified, each commodity uses its own commodity-specific transit times now. Since we will be looking at commodity-dependent transit times exclusively now, we will shorten the suffix "with commodity-dependent transit times" to "with cdtt" or even omit it completely for reasons of brevity from now on. We continue by adapting the network flow problems introduced in chapter 2 to the new model.

Maximum Multicommodity Flow over Time Problem with CDTT	
Instance:	A network $G = (V, A)$ with capacities u_a, a set of commodities $K = \{1, \dots, k\}$, transit times τ^i_a for every arc $a \in A$ and every commodity $i \in K$; sets of sources $S^i_+ \subseteq V$ and sets of sinks $S^i_- \subseteq V$ with $S^i_+ \cap S^i_- = \emptyset$ for every commodity $i \in K$; a time horizon $T \geq 0$.
Task:	Find a feasible multicommodity flow over time with cdtt f with time horizon T of maximum value $\lvert f \rvert$.

Multicommodity Transshipment over Time Problem with CDTT	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$, a set of commodities $K = \{1, \dots, k\}$, transit times τ^i_a for all arcs $a \in A$ and all commodities $i \in K$; supplies and demands b^i_v for every node $v \in V$ and every commodity $i \in K$; a time horizon $T \geq 0$.
Task:	Find a feasible multicommodity transshipment over time with cdtt f with time horizon T fulfilling all supplies and demands b.

Quickest Multicommodity Transshipment Problem with CDTT	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$, a set of commodities $K = \{1, \ldots, k\}$, transit times τ_a^i for all arcs $a \in A$ and all commodities $i \in K$; supplies and demands b_v^i for every node $v \in V$ and every commodity $i \in K$.
Task:	Find the minimum time horizon $T \geq 0$ such that a feasible multicommodity transshipment over time with cdtt f with time horizon T fulfilling all supplies and demands b exists.

Earliest Arrival Multicommodity Transshipment Problem with CDTT	
Instance:	A network $G = (V, A)$ with capacities u_a for every arc $a \in A$, a set of commodities $K = \{1, \ldots, k\}$, transit times τ_a^i for all arcs $a \in A$ and all commodities $i \in K$; supplies and demands b_v^i for every node $v \in V$ and every commodity $i \in K$.
Task:	Find the minimum time horizon $T \geq 0$ and a feasible multicommodity transshipment over time with cdtt f with time horizon T such that f fulfills all supplies and demands b and $\lvert f \rvert$ is maximal at every point in time $t \in [0, T]$.

Once again, the problems remain rather unchanged, at least as far as their input and their task is concerned. In addition to these problems we define Integer flow and transshipment problems. These problems consist of finding a multicommodity flow or transshipment over time f whose flow values f_a^i are integer for all $a \in A, i \in K$ and only change at discrete time steps $\theta \in \mathbb{N}_0$, i.e. $f_a^i \in N_0$. We call such a flow or transshipment *integer* and refer to the specific integer flow problems as Integer Maximum Multicommodity Flow over Time Problem and so on.

Commodity-dependent transit times obviously have the advantage of being able to model reality more closely, since agents or vehicles moving at different speeds through a common network can easily be modelled by them. Especially if flow represents agents or vehicles it is interesting to consider integer flows and transshipments, since agents or vehicles can usually not be split up arbitrarily. More realism is usually comes achieved at the cost of increased complexity - therefore we will now take a look at the complexity of the new problems.

4.2 Complexity

Normal transit times are obviously a special case of commodity-dependent transit times. Therefore, the new problems are at least as hard as the problems described in chapter 2 and 3.

Problem	Complexity
Transshipment over Time Problem	NP-hard
Minimum Cost Flow over Time Problem	NP-hard
Quickest Transshipment Problem	NP-hard
Earliest Arrival Transshipment Problem	NP-hard

Table 4.1: Complexity of multicommodity network problems with commodity-dependent transit times.

In fact, we can show that even severly restricted versions of these problems remain NP-hard. Consider multicommodity flow and transshipment problems where all commodities $i \in K$ have a single source s_i and a single sink t_i with $s_i = s_j$ and $t_i = t_j$ for all $i, j \in K$. This restriction causes multicommodity flow and transshipment problems with normal transit times to be reduced to single-commodity problems, essentially.

Before we show these reductions we make some helpful observations.

Observation 4.2.1. *Let $G = (V, A)$ be the network of some multicommodity flow or transshipment over time problem with a set of commodities K, capacities u, transit times τ and a time horizon T and $s, t \in V$. Let f be a solution to this problem, i.e. multicommodity flow or transshipment over time f. Then the following statements hold*

1. *If for some commodity $i \in K$ there is only a single s-t-path with $\tau^i(P) < T$ and $b^i_s = -b^i_t \neq 0$ then P must be used to send b^i_s units of flow of commodity i from s to t (assuming that $u_{min}(P)$ is large enough).*

2. *The total amount of flow entering an arc $a \in A$ can be bounded from above by*

$$\int_0^T f^i_a(\theta)\ d\theta \leq \max\left\{(T - \tau^i_a), 0\right\} \cdot u_a$$

   due to $f^i_a(\theta) = 0$ for $\theta \notin [0, T - \tau^i(\theta))$ and $f^i_a(\theta) \leq u_a$ for $\theta \in \mathbb{R}$.

Theorem 4.2.2. *The* Multicommodity Transshipment over Time Problem with CDDT *and the* Quickest Multicommodity Transshipment Problem with CDDT *remain NP-hard even if there is only a single source $s \in V$ and a single sink $t \in V$ for all commodities $i \in K$, i.e. $b^i_v = 0\ \forall i \in K, v \in V \setminus \{s, t\}$ and $b^i_s > 0, b^i_t < 0\ \forall i \in K$.*

Proof. We reduce the well-known NP-hard Partition Problem to the Multicommodity Transshipment over Time Problem with CDDT and the above mentioned restrictions. Consider an instance I of the partition problem consisting of n numbers $x_1, \dots, x_n$ with $\sum_{i=1}^{n} x_i = 2L$ for some $L \in \mathbb{N}$. The PARTITION PROBLEM asks whether a subset $J \subseteq \{1, \dots, n\}$ exists such that

$$\sum_{i \in J} x_i = L$$

holds. For such an instance I we define an instance $I' = (G = (V, A), K, u, \tau, b, T)$ of the MULTICOMMODITY TRANSSHIPMENT OVER TIME PROBLEM WITH CDDT. We will use $2n + 1$ commodities and unit capacities, i.e. $K = \{1, \dots, 2n + 1\}$ and $u_a = 1$ for all $a \in A$. For some $1 > \epsilon > 0$ we define the time horizon as $T := L + \epsilon$. We define a network $G = (V, A)$ by

$$\begin{aligned}
V &:= \{v_1, \dots, v_{n+1}, u_1, \dots, u_n, w_1, \dots, w_n\} \\
A &:= \bigcup_{i=1}^{2n+1} A_i \\
A_1 &:= \{ (v_j, u_j), (v_j, w_j), (u_j, v_{j+1}), (w_j, v_{j+1}) \mid j \in \{1, \dots, n\}\} \\
A_i &:= \begin{cases} \{(v_1, u_{i-1}), (v_i, t)\} & i \in \{2, \dots, n\} \\ \{(v_1, w_{i-n-1}), (v_{i-n}, t)\} & i \in \{n+2, \dots, 2n\} \end{cases}
\end{aligned}$$

and $A_{n+1} := \{(v_1, u_n)\}$, $A_{2n+1} := \{(v_1, w_n)\}$. There are no parallel arcs in A_1, therefore we can identify its arcs by their pair of nodes. For convenience, we define $s := v_1$ and $t := v_{n+1}$. Note that we consider arcs $(v, w) \in A_i$ and $(v, w) \in A_j$ to be different if $i \neq j$. We define the transit times, supplies and demands of the first commodity by

$$\tau_a^1 := \begin{cases} x_j & a = (v_j, u_j) \in A_1, j \in \{1, \dots, n\} \\ T & a \in A_i, i > 1 \\ 0 & \text{else} \end{cases} \qquad b_v^1 := \begin{cases} 2\epsilon & v = s \\ -2\epsilon & v = t \\ 0 & \text{else} \end{cases}$$

Figure 4.1 shows the subnetwork (V, A_1) and the transit times of the first commodity.

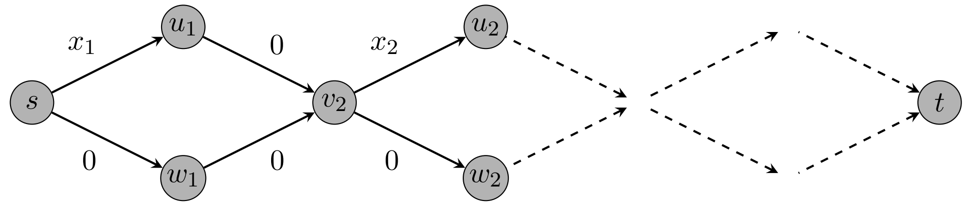


Figure 4.1: Network (V, A_1). Edge labels denote transit times of the first commodity.

The transit times, supplies and demands of the other $2n$ commodities $i = 2, \ldots, 2n+1$ are illustrated in figures 4.2 and 4.3. Formally, they are defined by:

$$\tau_a^i := \begin{cases} 0 & a = (u_{i-1}, v_i) \in A_1, i \in \{2, \ldots, n+1\} \\ 0 & a = (w_{i-1}, v_i) \in A_1, i \in \{n+2, \ldots, 2n+1\} \\ 0 & a \in A_i, i \in \{2, \ldots, 2n+1\} \\ T & \text{else} \end{cases} \qquad b_v^i := \begin{cases} L & v = s \\ -L & v = t \\ 0 & \text{else} \end{cases}$$

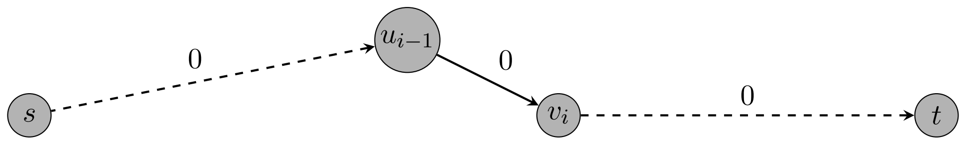


Figure 4.2: Network $(V, A_i \cup \{(u_{i-1}, v_i)\}$ for $i \in \{2, \ldots, n+1\}$. Edge labels denote transit times of commodity i; the dashed arcs are only usable by commodity i and all arcs not shown are not usable by commodity i due to their respective transit times. The non-dashed arc is $(u_{i-1}, v_i) \in A_1$.

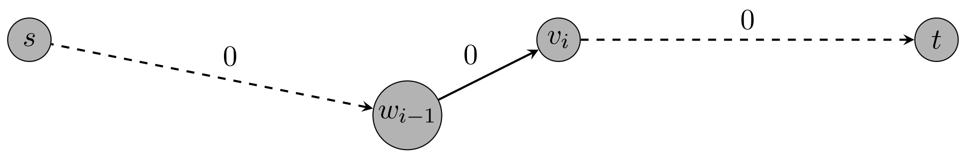


Figure 4.3: Network $(V, A_i \cup \{(w_{i-1}, v_i)\}$ for $i \in \{n+2, \ldots, 2n+1\}$. Edge labels denote transit times of commodity i; the dashed arcs are only usable by commodity i and all arcs not shown are not usable by commodity i due to their respective transit times. The non-dashed arc is $(w_{i-1}, v_i) \in A_1$.

By construction it is clear that the size of I' is polynomial in the size of I, i.e. number of values $x_1, \ldots, x_n$ given for the Partition instance. We have to show that there is a solution to I' if and only if there is a solution to I.

Assume that there is a solution to I, i.e. a subset $J \subseteq \{1, \ldots, n\}$ such that

$$\sum_{i \in J} x_i = L$$

holds. We construct a multicommodity transshipment over time f that is a solution to instance I in two steps. We begin by constructing two arc-disjoint s-t-paths P and Q along which we send exactly ϵ units of flow of the first commodity. This fulfills the supplies and demands of the first commodity. We construct P by using the arcs $(v_j, u_j), (u_j, v_{j+1}) \in A_1$ for all $j \in J$ and $(v_j, w_j), (w_j, v_{j+1}) \in A_1$ for all $j \notin J$; similiar we construct Q by using the arcs $(v_j, w_j), (w_j, v_{j+1}) \in A_1$ for all $j \in J$ and $(v_j, u_j), (u_j, v_{j+1}) \in A_1$ for all $j \notin J$. By definition of J it follows that $\tau^1(P) = \tau^1(Q) = L$ and $u_{min}(P) = u_{min}(Q) = 1$. Therefore it is possible to send 2ϵ units of flow of the first commodity from s to t using P and Q.

We continue by fulfilling the supplies and demands of the other commodities. By construction, there is only a single s-t-path P_i with $\tau^i(P_i) < T$ for all commoditities

$i \in \{2, \ldots, 2n+1\}$. All paths $P_i, P_j, i \neq j$ are arc-disjoint and share only a single arc with either P or Q. Due to $\tau^i(P_i) = 0$ and $u_{min}(P_i) = 1$ it is possible to send $T = L + \epsilon$ units of flow along P_i on the assumption that no other flow is sent through an arc contained in P_i. We know that exactly one arc of P_i is used by either P or Q to send ϵ units of flow of the first commodity through it. Therefore we can still send $T - \epsilon = L$ units of flow of commodity i, satisfying the supplies and demands of commodity i. This constitutes a solution to I'.

Now assume that f is a solution to I'. By construction, there is only a single s-t-path P_i with $\tau^i(P) < T$ for all commodities $i \in \{2, \ldots, 2n+1\}$. Note that two paths $P_i, P_j, i \neq j$ do not share any arcs and that any arc a of the form $a = (u_j, v_{j+1}) \in A_1$ or $a = (w_j, v_{j+1}) \in A_1$ for $j \in \{1, \ldots, n\}$ is used by exactly one path P_i. Since L units of flow of commodity i must pass through a, at most $T - L = \epsilon$ units of flow of other commodities can use a. By construction of the network it is also evident that 2ϵ units of flow of the first commodity must pass through each pair of arcs $(u_j, v_{j+1}), (w_j, v_{j+1}) \in A_1$, $j = 1, \ldots, n$. Therefore we can conclude that exactly ϵ units of flow are sent through each arc $(u_j, v_{j+1}), (w_j, v_{j+1}) \in A_1$, $j = 1, \ldots, n$ by f.

Any s-t-path P carrying flow of the first commodity must have length $\tau^1(P) < T = L + \epsilon$. We now assume that no $J \subseteq \{1, \ldots, n\}$ exist such that

$$\sum_{j \in J} x_j = L$$

holds. On this assumption, there cannot be an s-t-path P of length $\tau^1(P) = L$. We know that f sends 2ϵ units of flow of the first commodity from s to t along some s-t-paths. Let $\mathcal{P}$ be the set of these s-t-paths. Due to x_j, $j = 1, \ldots, n$ being integer we also know that $\tau^1(P)$ is integer. Let $x^1(P)$ denote the amount of flow of the first commodity sent along a path $P \in \mathcal{P}$. We write $v \in P$ or $a \in P$ if a node $v \in V$ or an arc $a \in A$ is contained in P. Consider the following weighted sum of path lengths. The length $\tau^1(P)$ of a path P depends solely on which arcs of the form $(v_j, u_j) \in A_1$ are contained in the path. This allows us to reorganize the sum as done below.

$$\begin{aligned}
\sum_{P \in \mathcal{P}} x^1(P) \cdot \tau^1(P) &= \sum_{P \in \mathcal{P}} x^1(P) \cdot \left(\sum_{j=1,\ldots,n: u_j \in P} a_j \right) \\
&= \sum_{j=1}^{n} a_j \cdot \left(\sum_{P \in \mathcal{P}: u_j \in P} x^1(P) \right)
\end{aligned}$$

We have already established that for an arc $a = (u_j, v_{j+1}) \in A_1$ the following holds:

$$\sum_{P \in \mathcal{P}: a \in P} x^1(P) = \epsilon$$

Thus, we can conclude that

$$\begin{aligned}\sum_{P\in\mathcal{P}} x^1(P)\cdot\tau^1(P) &= \sum_{j=1}^{n} a_j \cdot \left(\sum_{P\in\mathcal{P}:u_j\in P} x^1(P)\right)\\ &= \epsilon\sum_{j=1}^{n} a_j\\ &= 2\epsilon L\end{aligned}$$

We know now that $\tau^1(P)\in N_0$ and $\tau^1(P) < L$ for all $P\in\mathcal{P}$. It follows

$$\sum_{P\in\mathcal{P}} x^1(P)\cdot\tau^1(P) < \sum_{P\in\mathcal{P}} x^1(P)\cdot L = 2\epsilon L$$

in contradiction to the assumption. Therefore, a subset $J\subseteq\{1,\ldots,n\}$ with

$$\sum_{j\in J} x_j = L$$

must exist. This shows that the MULTICOMMODITY TRANSSHIPMENT OVER TIME PROBLEM WITH CDDT and the above mentioned restrictions is NP-hard as well and therefore the corresponding QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM as well.

□

We continue with the INTEGER QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT. We can reduce the SATISFIABILITY PROBLEM to the INTEGER QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT in a way that creates a gap between instances of the INTEGER QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT that correspond to instances of the SATISFIABILITY PROBLEM which have a variable assignment fulfilling all clauses and those that do not. The gap yields a non-approximability result for the INTEGER QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT.

Lemma 4.2.3. *For any instance I of the* SATISFIABLITY PROBLEM *we can construct an instance I' of the* INTEGER QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT *that is polynomial in the size of I and can be solved within a time horizon of 1 if and only if there is an assignment fulfilling all of the clauses of I. If there is no fulfilling assignment then I' cannot be solved within a time horizon smaller than 2.*

Proof. An instance I of the SATISFIABILITY PROBLEM consists of disjunctive clauses $c_1,\ldots,c_m$ over variables $x_1,\ldots,x_n\in\{0,1\}$. The task is to decide whether a variable assignment exists that fulfills all clauses. For such an instance I we construct an instance $I' = (G=(V,A),K,u,\tau,b)$ of the INTEGER QUICKEST MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT. We will use $n+m$ commodities and unit capacities, i.e. $K=\{1,\ldots,n+m\}$ and $u_a=1$ for all $a\in A$. For a clause c_j, $j=1,\ldots,m$ we write

$|c_j|$ for the number of literals in c_j. For a variable x_i, $i = 1, \dots, n$ we write $|x_i|$ for the number of occurences (whether negated or unnegated) of x_i in the clauses. Without loss of generality, we assume that $|x_i| \geq 1$ for $i = 1, \dots, n$.

We define a network $G_i = (V_i, A_i)$ with zero transit times for commodity i, supplies and demands for every variable x_i, $i = 1, \dots, n$. Let $k := |x_i| + 1$ for brevity.

$$V_i := \{s_i, t_i\} \cup \{\, y_{i,k}, n_{i,k} \mid k = 1, \dots, |x_i|\}$$
$$A_i := \{(s_i, y_{i,1}), (s_i, n_{i,1}), (y_{i,k}, t_i), (n_{i,k}, t_i)\} \cup \{\, (y_{i,j}, y_{i,j+1}), (n_{i,j}, n_{i,j+1}) \mid j \in \{1, \dots, k-1\}\}$$
$$b^i_v := \begin{cases} 1 & v = s_i \\ -1 & v = t_i \\ 0 & \text{else} \end{cases}$$

Figure 4.4 shows the subnetwork (V, A_i) for $i \in \{1, \dots, n\}$ and the transit times of commodity i.

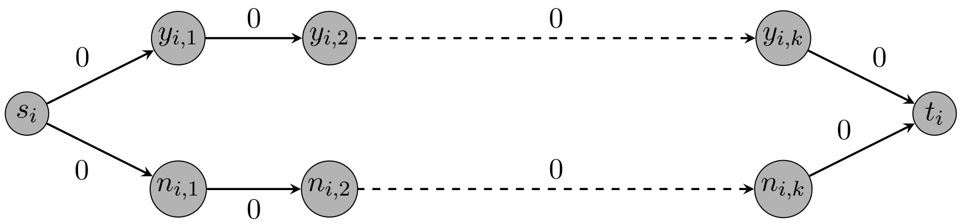


Figure 4.4: Network (V, A_i), $i = 1, \dots, n$. Edge labels denote transit times of commodity i.

We consider clause c_j, $j \in \{1, \dots, m\}$. We define a network $G_{n+j} = (V_{n+j}, A_{n+j})$ by $V_{n+j} := \{s_{n+j}, t_{n+j}\}$ and construct A_{n+j} iteratively by adding two arcs for every variable in c_j. If a positive literal of variable x_i is contained, we add $(s_{n+j}, n_{i,\#x_i}), (n_{i,\#x_i+1}, t_{n+j})$ where $\#x_i$ denotes the number of occurrences of x_i in $c_1, \dots, c_j$. We define $\tau^{n+j}_a = 0$ for $a = (n_{i,\#x_i}, n_{i,\#x_i+1}) \in A_i$. If a negative literal of x_i is contained, we add $(s_{n+j}, y_{i,\#x_i}), (y_{i,\#x_i+1}, t_{n+j})$. We define $\tau^{n+j}_a = 0$ for $a = (y_{i,\#x_i}, y_{i,\#x_i+1}) \in A_i$ and $\tau^{n+j}_a = 0$ for $a \in A_{n+j}$. Supplies and demands are defined as follows:

$$b^{n+j}_v := \begin{cases} 1 & v = s_{n+j} \\ -1 & v = t_{n+j} \\ 0 & \text{else} \end{cases}$$

We define the network $G = (V, A)$ by $V := \bigcup_{i=1}^{n+m} V_i$ and $A := \bigcup_{i=1}^{n+m} A_i$. We use unit capacities and extend supplies and demands as follows:

$$b^i_v := \begin{cases} 1 & v = s_i \\ -1 & v = t_i \\ 0 & \text{else} \end{cases}$$

All transit times $\tau_a^i, i \in K, a \in A$ that are not defined yet are set to a large constant T. $T = 2$ is sufficient for our purposes. There are no parallel arcs in A, therefore we can

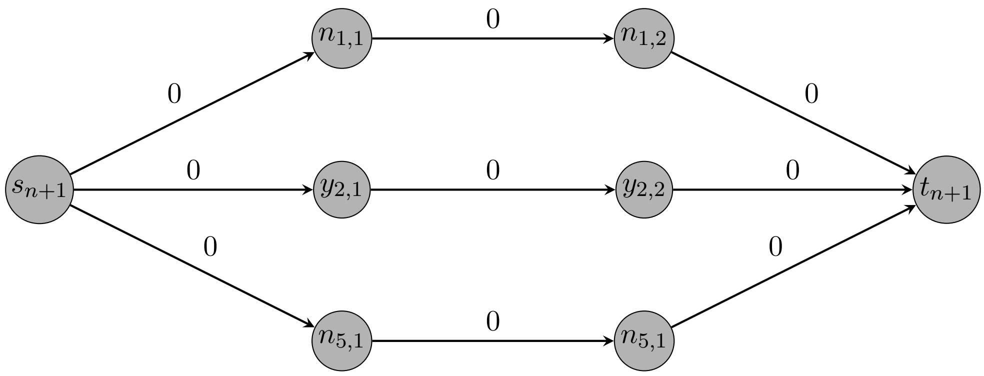


Figure 4.5: Network (V, A_{n+1}) if clause c_1 has the form $c_1 = x_1 \vee \overline{x_2} \vee x_5$. Edge labels denote transit times of commodity $n + 1$, edges not usable by commodity $i + 1$ are not shown.

identify its arcs by their pair of nodes. It is obvious that the size of I' is polynomial in the size of I.

Assume there is a variable assignment $x_1, \ldots, x_n$ fulfilling all clauses of I. Then there is a solution to I' as well that requires only a time horizon of 1. For every of the commodities $i = 1, \ldots, n$ which correspond to the variables x_i of the Satisfiability Problem there are only two two possible s_i-t_i-paths to send the single flow unit of the commodity. If $x_i = 1$ in the assignment, we send the flow along the path containing $y_{i,1}$, otherwise we send it along the path containing $n_{i,1}$.

For every clause c_j, $j = 1, \ldots, m$ there is at least one literal that is fulfilled. Assume without loss of generality that x_k is such a (positive) literal for c_j. We send the flow unit of the commodity $n + j$ which corresponds to the clause c_j along the s_{n+j}-t_{n+j}-path containing $n_{k,\#x_k}$. Since we are sending whole flow units along paths of length 0 for their respective commodities, it is obvious that the flow induced by these paths is a solution to I' if the capacity constraints are fulfilled. Each commodity $i \in \{1, \ldots, n\}$ uses only arcs of A_i and every commodity $j \in \{n + 1, \ldots, n + m\}$ uses only arcs of A_j and A_i for some fulfilled literal x_i in c_{j-n}. In particular, there are at most 2 commodities that can use any arc $a \in A$ by construction. Thus, a violation of the capacity constraints is only possible between a commodity $i \in \{1, \ldots, n\}$ and a commodity $j \in \{n + 1, \ldots, n + m\}$ if clause c_{j-n} contains a fulfilled literal x_i which is used to determine the path along which the flow unit of commodity j is sent. However, we have chosen the paths for flow of commodity i and j in a way that guarantees that the path for j uses only n-nodes if the path for i uses y-nodes and vice versa. Therefore we have a solution with a time horizon of 1. It is obvious that we cannot send the flow any faster due to the integrality constraints.

Now we assume that there is a transshipment over time f that solves I' with time horizon 1. Thus, there are s_i-t_i-paths of length 0 for every commodity $i = 1, \ldots, n + m$.

Due to the integrality constraints there is a single s_i-t_i-path P_i for every commodity i. For $i = 1, \ldots, n$ that means that either the y-nodes or the n-nodes are contained in P_i. In the first case, we define $x_i := 1$ and $x_i := 0$ in the other. We know that for every commodity $i \in \{n+1, \ldots, n+m\}$ there is a s_i-t_i-path P_i along which one unit of flow from this commodity is sent within a time horizon of 1. If P_i is of the form $\{(s_i, y_{k,l}), (y_{k,l}, y_{k,l+1}), (y_{k,l+1}, t_i)\}$ for some $k, l \in \mathbb{N}_0$ it follows that $\overline{x_k}$ is contained in clause c_{i-n}. In the other case, if P_i is of the form $\{(s_i, n_{k,l}), (n_{k,l}, n_{k,l+1}), (n_{k,l+1}, t_i)\}$ for some $k, l \in \mathbb{N}_0$ it follows that x_k is contained in clause c_{i-n}. Thus, if commodity $j \in \{n+1, n+m\}$ sends flow along a path containing y-nodes of V_k within time horizon 1 it follows that commodity k must use the path containing only n-nodes of V_k for its own flow unit. This corresponds to setting $x_k := 0$ and to a clause c_{j-n} containing $\overline{x_k}$, i.e. a fulfilled clause c_{j-n}. Similiar it follows if commodity j uses n-nodes of $V - k$ and commodity k uses y-nodes that c_j is fulfilled.

Therefore we have shown that a variable assignment fulfilling all clauses of I exists if and only if a transshipment over time exists that solves I' within a time horizon of 1. Due to the integrality restrictions it is obvious that no transshipment over time with a smaller time horizon exists for I'. If no transshipment over time with time horizon 1 exists that solves I' it once again follows from the integrality constraints that any solution must have a time horizon of at least 2.

□

With the gap that we have shown in this lemma, we can formulate a non-approximability result for the Integer Quickest Multicommodity Transshipment Problem with CDDT.

Theorem 4.2.4. *Unless $P = NP$, there is no polynomial time approximation scheme for the* Integer Quickest Multicommodity Transshipment Problem with CDDT.

Proof. The proof of the theorem follows directly from the previous lemma. □

We conclude this section with this result and complete the chapter by examining two aspects of the commodity-dependent transit times that we have not considered yete.

4.3 Overtaking and Transit-Time-Restrictions

The first aspect we will take into account is *overtaking*. This refers to the phenomenon that flow enters an arc a at time t and leaves it at time $t + \delta$ whereas another flow enters a at time $t' > t$ and leaves at $t' + \delta' < t + \delta$. Thus, the first flow is overtaken by the latter flow. If overtaking is possible, we also say that the *first-in-first-out property* is violated. It is rather obvious that the first-in-first-out property does not hold for our setting of commodity-dependent transit times. In some applications, this is an unwanted phenomenon - think for example of a railroad network; trains can usually not overtake each other arbitrarily. There are other applications where overtaking is common, e.g. on highways. Thus, the question of whether overtaking is a problem is rather application specific. We have seen in the last chapter that problems with arbitrary transit times are not easy to solve, especially when integrality restrictions are added. When dealing with a concrete application, it could therefore be useful to consider transit times that are not arbitrary, but somehow restricted. One possible way to accomplish this would be to assign a length l_a to every arc $a \in A$ and a velocity v_i to every commodity $i \in K$; commodity-dependent transit times can then be defined by

$$\tau_a^i := \frac{l_a}{v_i} \quad \forall a \in A, i \in K$$

Such restricted transit times would not be very unrealistic in some cases, e.g. if we are considering vehicles moving through a road network. We will not, however, examine the benefits of restricted transit times or the problems of overtaking in further detail in this thesis due to the sometimes very application specific aspects and the scope-constraints of this thesis. Both aspects would be interesting topics for further research, though.

Chapter 5

Quickest Transshipments

In this chapter, we will consider the quickest transshipment problem with multiple commodities and commodity-dependent transit times. We have already seen that the Quickest Multicommodity Transshipment Problem with CDDT is NP-hard, so we cannot expect to find an exact, efficient algorithm to solve it. Thus, we will focus on developing a fully polynomial time approximation scheme for this problem. We will use a discrete time model in this chapter. Note that we allow non-integer flow values as we have already seen that unless $P = NP$ no FPTAS exists for the Integer version of this problem.

5.1 FPTAS

In order to develop an FPTAS, we require some tools which we will introduce now. The first of these tools are condensed time-expanded networks. These tools and many of the techniques employed in this chapter are due to Fleischer and Skutella [7, 8].

Definition 5.1.1 (Condensed Time-Expanded Networks)**.** *Consider a network $G = (V, A)$, capacities $u_a, a \in A$, transit times $\tau_a \in \mathbb{N}_0, a \in A$ and a time horizon $T \in \mathbb{N}$. We can construct the time-expanded network G^T for G as defined in chapter 2. The disadvantage of using G^T is that its size depends linearly on T and is only pseudopolynomial in the input-size. If T and all τ_a are multiples of some $\Delta \geq 1$ we can define a Δ-time expanded network G^T_Δ by dividing all transit times and the time horizon by Δ. If Δ is large enough such that $\frac{T}{\Delta}$ is polynomially bounded in the input size then the size of G^T_Δ is polynomially bounded in the input size as well. Every arc in the Δ-condensed network corresponds to an arc and a time-interval Δ. Therefore, we have to scale capacities by a factor of Δ. Let $\overline{u}$, $\overline{\tau}$ and $\overline{T}$ be the capacities, transit times and time horizon for the condensed network, respectively. They are defined by*

$$\overline{u}_a := \Delta \cdot u_a \forall a \in A, \quad \overline{\tau}_a := \frac{\tau_a}{\Delta} \quad \forall a \in A, \quad \overline{T}^i_a := \frac{T}{\Delta}$$

The Δ-condensed time-expanded network $G_\Delta^T = (V_\Delta^T, A_\Delta^T)$ is given by

$$\begin{aligned}
V_\Delta^T &:= \left\{ v_t \mid v \in V, t \in \left\{0, 1, \ldots, \overline{T}\right\}\right\} \\
A_\Delta^T &:= A_1 \cup A_2 \\
A_1 &:= \left\{ a_t = (v_t, w_{t+\overline{\tau}_a}) \mid a = (v, w) \in A, t \in \left\{0, \ldots, \overline{T} - \overline{\tau}_a\right\}\right\} \\
A_2 &:= \left\{ a_{v,t} = (v_t, v_{t+1}) \mid v \in V, t \in \left\{0, \ldots, \overline{T} - 1\right\}\right\}
\end{aligned}$$

Note that the definition above is for commodity-independent transit times only. We can extend this definition to commodity-dependent transit times by adding copies of every arc for every commodity. We assume that τ_a^i is a multiple of Δ for $i \in K$, $a \in A$ and define

$$\overline{\tau}_a^i := \frac{\tau_a^i}{\Delta} \quad \forall a \in A, i \in K$$

An arc of the form $a_t = (v_t, w_{t+\overline{\tau}_a})$ in the previous definition is replaced by $|K|$ arcs $a_{t,i} = (v_t, w_{t+\overline{\tau}_a^i})$ with $i \in K$. Using the Δ-condensed time-expanded network in this setting is more difficult - only commodity i may use arcs $a_{t,i}$ which must be enforced by algorithms using the Δ-condensed time-expanded network. The following theorems and lemmas will assume that static transshipments in the time-expanded network fulfill this property. As we have seen for normal time-expanded networks, there is an equivalence between static transshipments in the Δ-condensed time-expanded network and transshipments over time in the original network.

Lemma 5.1.2. *If T and τ_a^i are multiples of Δ for all $i \in K, a \in A$, then every multicommodity transshipment over time f in a network G that is completed within the time horizon T corresponds to a static multicommodity transshipment x in the Δ-condensed time-expanded network G_Δ^T and every static multicommodity transshipment in G_Δ^T corresponds to a multicommodity transshipment over time that is completed within the time horizon T.*

Proof. Let f be an arbitrary multicommodity transshipment over time in G. We define a static multicommodity transshipment x by averaging the flow values $f_a^i(\theta), \theta \in [p\Delta, (p+1)\Delta), p = 0, \ldots, T-1$ for every $i \in K, a \in A$. The averaged value is then assigned to the corresponding arc in the Δ-condensed time-expanded network. Note that the overall amount of flow sent does not change and is bounded by Δu_a which guarantees that the capacity contraints are fulfilled. We define the flow values of the holdover arcs such that the flow conservation contraints are fulfilled (which is possible due to f being a transshipment over time).

On the other hand we can construct a multicommodity transshipment over time f in G from a static multicommodity transshipment x in G_Δ^T. If $\overline{a} \in A_\Delta^T$ is an arc corresponding to a time interval $[p\Delta, (p+1)\Delta), p = 0, \ldots, T-1$ and an arc $a \in A$ then we set $f_a^i(\theta) := \frac{x_{\overline{a}}}{\Delta}$ for $\theta \in [p\Delta, (p+1)\Delta)$. Due to the choice of our capacities we are guaranteed that the flow values of f are feasible and the total amount of flow sent is not changed either. □

We can state a similar, but slightly weaker result if T is no multiple of Δ.

Corollary 5.1.3. *If τ_a^i are multiples of Δ for all $i \in K, a \in A$, then every multicommodity transshipment over time f in a network G that is completed within a time horizon T corresponds to a static multicommodity transshipment x in the Δ-condensed time-expanded network G_Δ^T and every static multicommodity transshipment in G_Δ^T corresponds to a multicommodity transshipment over time that is completed within the time horizon $T + \Delta$.*

When using integer transit times then we can obviously find a Δ such that all transit times are a multiple of Δ by choosing $\Delta := 1$. This choice of Δ does not reduce the size of the network, of course. Therefore we have to consider the case of transit times which are not multiples of Δ. In this case, we introduce new transit times which are rounded up, i.e. $\overline{\tau}_a^i := \left\lceil \frac{\tau_a^i}{\Delta} \right\rceil \Delta$. It follows that $0 \leq \overline{\tau}_a^i - \tau_a^i \leq \Delta$ for $\forall i \in K, a \in A$.

If we choose Δ large enough then G_Δ^T will be polynomial in the input size of the instance. Then we can solve a static multicommodity transshipment problem on the Δ-condensed time expanded network by solving a linear program. The question is now how much accuracy is lost due to the rounding.

We need the following lemma in order to answer this question.

Lemma 5.1.4. *Let T be the optimal time horizon for an instance of the* Quickest Multicommodity Transshipment Problem with CDDT *with supplies and demands b and integer transit times. For any $\delta \geq 1$ there is a transshipment over time f that fulfills supplies and demands δb, i.e. supplies and demands scaled by a factor of δ, within a time horizon of δT.*

Proof. Consider an optimal solution f^* to the instance of the Quickest Multicommodity Transshipment Problem with CDDT and the corresponding static multicommodity transshipment x in the time-expanded network. Now consider a modified version of the instance where all transit times and the time horizon are multiplied by a factor of δ. The δ-condensed time-expanded network of the modified instance is equivalent to the time-expanded network of the original instance, but with capacities that are larger by a factor of δ. We can scale the flow values of the static multicommodity transshipment x^* by a factor of δ and obtain a static multicommodity transshipment δx^* that is feasible in the δ-condensed time-expanded network of the modified instance and fulfills supplies and demands δb. As we have seen in the previous lemma, δx^* corresponds to a transshipment over time with a time horizon of δT in the original network, which concludes our proof. □

We now set $\Delta := \frac{\epsilon^2 T}{n}$ and $\overline{T} := \lceil (1+\epsilon)^3 \frac{T}{\Delta} \rceil \Delta$. For this choice of Δ, the number of arcs in the Δ-condensed time-expanded network is $O(\frac{mnk}{\epsilon^2})$ which is polynomial in the input size and ϵ^{-1}.

Lemma 5.1.5. *For the choices of Δ and $\overline{T}$ as above there is a static multicommodity transshipment x in the Δ-condensed time-expanded network $G_\Delta^{\overline{T}}$ that fulfills supplies and demands $(1+\epsilon)b$.*

Proof. According to the previous lemma it is sufficient to show that in the network G with rounded transit times $\overline{\tau}$ a multicommodity transshipment over time exists that fulfills supplies and demands $(1+\epsilon)b$ within a time horizon of $\overline{T}$.

We now consider a multicommodity transshipment over time f as in the previous lemma with $\delta = (1+\epsilon)^2$. f can be decomposed into simple paths with delays $\mathcal{P}^d$ which consist of a sequence of nodes $(v_0, \ldots, v_k)$ and delays $(d_0, dots, d_k)$. The number d_i specifies how long flow waits in node v_i before travelling onward.

Then the total flow of a commodity $i \in K$ entering an arc $a \in A$ at time θ is

$$f_a^i(\theta) = \sum_{P^d \in \mathcal{P}^\lceil : a \in P} f_{P^d}^i(\theta - \tau(P^d, a))$$

with

$$\tau^i(P^d, a = (v_l, v_{l+1})) := \sum_{j=1}^{l} (\tau_a^i + d_j)$$

Based on f we construct a smoothed multicommodity transshipment f' using the decomposition of f into paths with delays by defining

$$f'^i_{P^d}(\theta) := \frac{1}{\epsilon T} \int_{\theta - \epsilon T}^{\theta} f_{P^d}^i(\xi)\ d\xi$$

for all $i \in K$, $\theta \in [0, (1+\epsilon)^2 T + \epsilon T)$ and $P^d \in \mathcal{P}^d$. It is easy to verify that f' is still feasible and sends the same amount of flow on path P^d thus fulfilling the supplies and demands $(1+\epsilon)^2 b$ as well.

We can send flow according to these smoothed paths in the network with the rounded transit times $\overline{\tau}$ as well but we are not guaranteed that it is feasible in this setting. Since every path $P^d \in \mathcal{P}^d$ is simple, it contains at most $n-1$ nodes. Thus,

$$0 \le \overline{\tau}^i(P^d) - \tau^i(P^d) \le \epsilon^2 T$$
$$0 \le \overline{\tau}^i(P^d, a) - \tau^i(P^d, a) \le \epsilon^2 T \qquad \forall a \in P^d$$

We will now examine to which extent the capacity constraints are violated when using f' with the rounded transit times $\overline{\tau}$. For an arc $a \in A$, a commodity $i \in K$ and $\theta \in [0, (1+\epsilon)^2 T + \epsilon T + \epsilon^2 T)$ we get

$$\begin{aligned} f'^i_a(\theta) &= \sum_{P^d \in \mathcal{P}^\lceil : a \in P} f'^i_{P^d}(\theta - \overline{\tau}^i(P^d, a)) \\ &= \frac{1}{\epsilon T} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \int_{\theta - \overline{\tau}^i(P^d, a) - \epsilon T}^{\theta - \overline{\tau}^i(P^d, a)} f_{P^d}^i(\xi)\ d\xi \end{aligned}$$

We now refer to the result that a path is at most $\epsilon^2 T$ longer when using the rounded transit times $\overline{\tau}$:

$$\frac{1}{\epsilon T} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \int_{\theta - \overline{\tau}^i(P^d,a) - \epsilon T}^{\theta - \overline{\tau}^i(P^d,a)} f^i_{P^d}(\xi)\ d\xi \leq \frac{1}{\epsilon T} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \int_{\theta - \tau^i(P^d,a) - \epsilon T - \epsilon^2 T}^{\theta - \tau^i(P^d,a)} f^i_{P^d}(\xi)\ d\xi$$
$$= \frac{1}{\epsilon T} \int_{\theta - \epsilon T - \epsilon^2 T}^{\theta} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} f^i_{P^d}(\xi - \tau^i(P^d, a))\ d\xi$$

In order to obtain a bound for the amount of flow of commodity i entering arc a at time θ, we transform the path based formulation into an arc based formulation:

$$\frac{1}{\epsilon T} \int_{\theta - \epsilon T - \epsilon^2 T}^{\theta} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} f^i_{P^d}(\xi - \tau^i(P^d, a))\ d\xi = \frac{1}{\epsilon T} \int_{\theta - \epsilon T - \epsilon^2 T}^{\theta} f^i_a(\xi)\ d\xi$$

Due to f being feasible it follows that

$$\frac{1}{\epsilon T} \int_{\theta - \epsilon T - \epsilon^2 T}^{\theta} f^i_a(\xi)\ d\xi \leq \frac{\epsilon^2 T + \epsilon T}{\epsilon T} u_a = (1+\epsilon) u_a$$

yielding the desired bound.

Thus, scaling f' by a factor of $\frac{1}{1+\epsilon}$ leads to a transshipment over time $\overline{f}$ that is feasible when using the rounded transit times $\overline{a}$. Since f fulfills the supplies and demands $(1+\epsilon)^2 b$, $\overline{f}$ fulfills the supplies and demands $(1+\epsilon)b$. Note that $\overline{f}$ is completed within a time horizon of $(1+\epsilon)^2 T + \epsilon^2 T + \epsilon T \leq (1+\epsilon^3)T \leq \overline{T}$. Therefore a static multicommodity transshipment x exists in the Δ-condensed time-expanded network $G^{\overline{T}}_{\Delta}$ that fulfills supplies and demands $(1+\epsilon)b$.

□

We can use this lemma to obtain a transshipment for the original network as follows.

Lemma 5.1.6. *A static multicommodity transshipment x in the Δ-condensed time-expanded network $G^{\overline{T}}_{\Delta}$ that fulfills supplies and demands $(1+\epsilon)b$ (as in the previous lemma) can be used to construct a multicommodity transshipment over time f in the original network G that fulfills all supplies and demands b within a time horizon of $(1+\epsilon)\overline{T}$.*

Proof. Given such a static multicommodity transshipment x in the Δ-condensed time-expanded network $G^{\overline{T}}_{\Delta}$ we construct the corresponding transshipment over time $\overline{f}$ with rounded up transit times $\overline{\tau}$ and a path decomposition $f^d_P, P^d \in \mathcal{P}^d$.

We construct a smoothed multicommodity transshipment f' using the decomposition of $\overline{f}$ into paths with delays by defining

$$f'^i_{P^d}(\theta) := \frac{1}{\epsilon \overline{T}} \int_{\theta - \epsilon \overline{T}}^{\theta} \overline{f}^i_{P^d}(\xi)\ d\xi$$

for all $i \in K$, $\theta \in [0, (1+\epsilon)\overline{T})$ and $P^d \in \mathcal{P}^d$. f' can be interpreted as a multicommodity transshipment over time with time horizon $(1+\epsilon)\overline{T}$ that satisfies supplies and demands

$(1+\epsilon)b$ in the setting with the original transit times τ but it is not guaranteed to be feasible in this context.

Once again, we will examine to which extent the capacity constraints are violated when using f' with the original transit times τ. For an arc $a \in A$, a commodity $i \in K$ and $\theta \in [0, (1+\epsilon)\overline{T})$ we get

$$\begin{aligned} f'^i_a(\theta) &= \sum_{P^d \in \mathcal{P}^\lceil : a \in P} f'^i_{P^d}(\theta - \tau^i(P^d, a)) \\ &= \frac{1}{\epsilon\overline{T}} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \int_{\theta - \tau^i(P^d,a) - \epsilon T}^{\theta - \tau^i(P^d,a)} \overline{f}^i_{P^d}(\xi)\, d\xi \end{aligned}$$

Again we now use the result that a path is at most $\epsilon^2 T$ longer when using the rounded transit times $\overline{\tau}$:

$$\begin{aligned} \frac{1}{\epsilon\overline{T}} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \int_{\theta - \tau^i(P^d,a) - \epsilon T}^{\theta - \tau^i(P^d,a)} \overline{f}^i_{P^d}(\xi)\, d\xi &\leq \frac{1}{\epsilon\overline{T}} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \int_{\theta - \overline{\tau}^i(P^d,a) - \epsilon \overline{T}}^{\theta - \overline{\tau}^i(P^d,a) + \epsilon^2 \overline{T}} \overline{f}^i_{P^d}(\xi)\, d\xi \\ &= \frac{1}{\epsilon\overline{T}} \int_{\theta - \epsilon\overline{T}}^{\theta + \epsilon^2 \overline{T}} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \overline{f}^i_{P^d}(\xi - \overline{\tau}^i(P^d, a))\, d\xi \end{aligned}$$

In order to obtain a bound for the amount of flow of commodity i entering arc a at time θ, we transform the path based formulation into an arc based formulation:

$$\frac{1}{\epsilon\overline{T}} \int_{\theta - \epsilon\overline{T}}^{\theta + \epsilon^2 \overline{T}} \sum_{P^d \in \mathcal{P}^\lceil : a \in P} \overline{f}^i_{P^d}(\xi - \overline{\tau}^i(P^d, a))\, d\xi = \frac{1}{\epsilon\overline{T}} \int_{\theta - \epsilon\overline{T}}^{\theta + \epsilon^2 \overline{T}} \overline{f}^i_a(\xi)\, d\xi$$

Due to f being feasible it follows that

$$\frac{1}{\epsilon\overline{T}} \int_{\theta - \epsilon\overline{T}}^{\theta + \epsilon^2 \overline{T}} \overline{f}^i_a(\xi)\, d\xi \leq \frac{\epsilon^2\overline{T} + \epsilon\overline{T}}{\epsilon\overline{T}} u_a = (1+\epsilon) u_a$$

yielding the desired bound. Scaling $\overline{f}$ by a factor of $\frac{1}{1+\epsilon}$ results in the required transshipment over time f. □

We consolidate our results into an FPTAS that consists mainly of a subroutine that determines for a given time horizon T and a precision $\epsilon > 0$ whether a feasible solution with a time horizon $(1+O(\epsilon))T$ for the corresponding Multicommodity Transshipment over Time Problem with CDDT exists. The FPTAS uses this subroutine embedded in a binary search framework. A binary search strategy can be employed to find lower and upper bounds for the optimal time horizon which differ only in a constant factor from each other with $\log T^{opt}$ calls to the subroutine, if T^{opt} is the optimal time horizon. Geometric mean binary search based on these bounds yields with $O(\log \epsilon{-}1)$ calls a time horizon $T \in [T^{opt}, (1+O(\epsilon))T^{opt}]$. The last call yields a solution with a time horizon bounded from above by $(1+\epsilon)\overline{T} \leq (1+\epsilon)^4 T + (1+\epsilon)\Delta$ and therefore in $(1+O(\epsilon))T^{opt}$.

Algorithm 1 FPTAS Subroutine

Input: Network G, commodities K, capacities u, commodity-dependent transit times τ, supplies and demands b, time horizon T, precision $\epsilon > 0$.

Output: A feasible multicommodity transshipment f fulfilling supplies and demands b within a time horizon of $(1+O(\epsilon))T$ or the information that supplies and demands cannot be fulfilled within time T.

1. Set $\Delta := \frac{\epsilon^2 T}{n}$ and $\overline{T} := \lceil (1+\epsilon)^3 \frac{T}{\Delta} \rceil \Delta$;
2. Compute a static multicommodity transshipment x in the Δ-condensed time-expanded network $G_\Delta^{overlineT}$ fulfilling supplies and demands $(1+\epsilon)b$. If this is not possible, output that the time horizon is too small.
3. Transform x into a multicommodity transshipment over time f in G with time horizon $(1+\epsilon)\overline{T}$ fulfilling supplies and demands b.

The Δ-condensed time-expanded network contains $O(\frac{n^2}{\epsilon^2})$ nodes and $O(\frac{nmk}{\epsilon^2})$ arcs. The static multicommodity transshipment in step 2 can therefore be computed by a linear program that is polynomial in the input size and ϵ^{-1}. Note that it is easy for us to guarantee that every commodity uses only the arcs with its own transit times since we can simply add constraints of the form $x_{a,i} = 0$ for all combinations of arcs $a \in A$ and commodities $i \in K$ that are not allowed. In order to execute step 3 in polynomial size we need to ensure that f can be encoded in a polynomial size. The following lemma shows this.

Lemma 5.1.7. *Consider the static transshipment x that is used to construct f in step 3 of the FPTAS subroutine. A path decomposition of x into flows on j paths in the condensed time-expanded network can be turned into a path decomposition of f on j paths such that each f_P is stepwise constant with $O(\frac{\overline{T}}{\Delta}) = O(n\epsilon^{-2})$ steps.*

Proof. Any path P used in a path-decomposition of x induces a path-flow over time $\overline{f}_P^d$ on a path with delays P^d such that the flow function $\overline{f}_P^d$ determining the flow entering the path P^d is 0 except for an interval of length Δ. Multiple paths in the condensed time-expanded network can correspond to the same path in the original network, possibly resulting in a flow function that is stepwise constant with as many non-zero steps as there are time layers in the condensed time-expanded network, i.e. $\frac{\overline{T}}{\Delta}$, which concludes the proof. □

Therefore the $f_P^i, i \in K$ have at most $O(n\epsilon^2)$ breakpoints as well and are piecewise linear due to

$$f_P^i(\theta) = \frac{1}{1+\epsilon}\frac{1}{\epsilon\overline{T}}\int_{\theta-\epsilon\overline{T}}^{\theta} \overline{f}_P^i(\xi)\ d\xi$$

and can be computed efficiently.

This leads to the following result.

Theorem 5.1.8. *For any instance of the* Quickest Multicommodity Transshipment Problem with CDDT *a* $(1+\epsilon)$*-approximate solution can be found for any* $\epsilon > 0$ *by* $O(\log \epsilon^{-1} + \log T^{opt})$ *static multicommodity transshipment computations in a condensed time-expanded network with* $O(n^2\epsilon^{-2})$ *nodes and* $O(nmk\epsilon^{-2})$ *arcs.*

We conclude the chapter with this result and continue with the Earliest Arrival Multicommodity Transshipment Problem with CDDT.

Chapter 6

Earliest Arrival Transshipments

In this chapter, we will examine earliest arrival transshipments in the context of multicommodity flows with commodity-dependent transit times. We already mentioned that earliest arrival transshipments are of special interest for evacuation planning and optimization, since they exactly embody the properties that are desirable for a perfect evacuation. With this application in mind, we focus mostly on INTEGER problems in this chapter.

6.1 Existence

The first question that we have to deal with is whether always an earliest arrival transshipment exists and if not, on which conditions an earliest arrival transshipment exists. We already know that an earliest arrival transshipment is not guaranteed to exist in the case of a single commodity and two sinks. This leads directly to the multicommodity case - if there are more than two sink nodes (between all commodities) then the existence of an earliest arrival transshipment is endangered. We know according to Gale, Richardson and Tardos [12, 25] that earliest arrival transshiments always exist in the case of multiple sources and a single sink. Note that if all commodities share the same sink node we essentially deal with a single-commodity problem; we can simply combine the commodities since they are sending flow to the same destination anyway. If we allow commodity-dependent transit times this is no longer true, however.

Theorem 6.1.1. *There are instances $I = (G = (V, A), K, b, \tau, u)$ of the* EARLIEST ARRIVAL MULTICOMMODITY TRANSSHIPMENT PROBLEM WITH CDDT *that have no solution, even if the network $G = (V, A)$ has only a single source s, a single sink t and one arc connecting both, i.e. $V = \{s, t\}, A = \{(s, t)\}$.*

Proof. We show the existence by defining such an instance $I = (G = (V, A), K, b, \tau, u)$ by $V := \{s, t\}$, $A := \{a = (s, t)\}$, $K := \{1, 2\}$, $u_a := 1$, $b_s^1 = b_2^2 = 1$, $b_t^1 = b_t^2 = -1$ and $\tau_a^1 = 1$, $\tau_a^2 = 2$. Figure 6.1 depicts the instance.

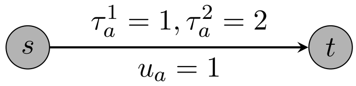


Figure 6.1: Network (V, A).

We need to send one unit flow of each commodity, which leaves us with only two options. We either send the unit of commodity 1 first or the unit of commodity 2. Doing the latter will result in both units beginning to arrive by time 2 at the sink. Doing the former causes the first unit to begin arriving by time 1 and the second by time 3. An earliest arrival transshipment would need to have at least one unit of flow beginning to arrive by time 1 and an additional unit by time 2, which is not possible. □

Thus we cannot hope to find earliest arrival transshipments in the case of commodity-dependent transit times in general. As we have seen in the last theorem, even very small networks are not guaranteed to have an earliest arrival transshipment. This makes it quite unlikely to find interesting families of instances for which earliest arrival transshipments exist. We can neither hope to find a criterion for the existence of earliest arrival transshiments that relies solely on local structures. Let for example $G = (V, A)$ be the network of an instance of the Earliest Arrival Transshipment Problem for which no earliest arrival transshipment exists and let G be of the structure that is shown in figure 6.2: there are two sinks t_1, t_2 with exactly one incoming arc $a_1 = (v_1, t_1), a_2 = (v_2, t_2)$ per sink, respectively, for some $v_1, v_2 \in V$.

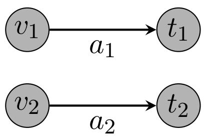


Figure 6.2: Network (V, A).

If we add an additional commodity with very large supplies that has v_1, v_2 as sources and t_1, t_2 as sinks then it is possible that the new instance has an earliest arrival transshipment - the arcs a_1, a_2 are saturated by flow of the new commodity for such a long time that it might not matter what happens within the rest of the instance. Therefore it is more viable to choose a different path to deal with these problems; this path is approximating earliest arrival transshipments.

6.2 Approximation

The following notion of approximating earliest arrival transshipments is due to Baumann and Koehler [2]. It relies on defining an α-earliest arrival transshipment, which we will do now as well.

Definition 6.2.1 (α-Earliest Arrival Transshipment)**.** *An α-earliest arrival transshipment is a transshipment over time f that fulfills at every point in time $t \in \mathbb{R}^+$ at least as many supplies and demands as possible in time $\frac{t}{\alpha}$, i.e. $|f(t)| \geq |f_{max}\left(\frac{t}{\alpha}\right)|$ if $|f_{max}(t)|$ denotes the maximal value possible for time t.*

Speaking figuratively, the flow of an α-earliest arrival transshipment f arrives at most α times later than flow of earliest arrival transshipment (if such a transshipment existed). We will now examine for which values of α we can compute α-earliest arrival transshipments for instances of the Earliest Arrival Multicommodity Transshipment Problem with CDDT.

Theorem 6.2.2. *There are instances of the* Earliest Arrival Multicommodity Transshipment Problem with CDDT *with a single source s and a single sink t for which no $\frac{3}{2}$-earliest arrival transshipment exists.*

Proof. Consider the following instance $I = (G = (V, A), K, b, \tau, u)$ of the Earliest Arrival Multicommodity Transshipment Problem with CDDT defined by $V := \{s, t\}$, $A := \{a = (s, t)\}$, $K := \{1, 2\}$, $u_a := 1$, $b_s^1 = b_2^2 = 1$, $b_t^1 = b_t^2 = -1$ and $\tau_a^1 = 0$, $\tau_a^2 = 1$. Figure 6.3 depicts the instance.

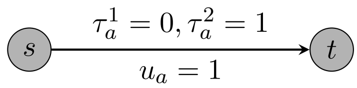
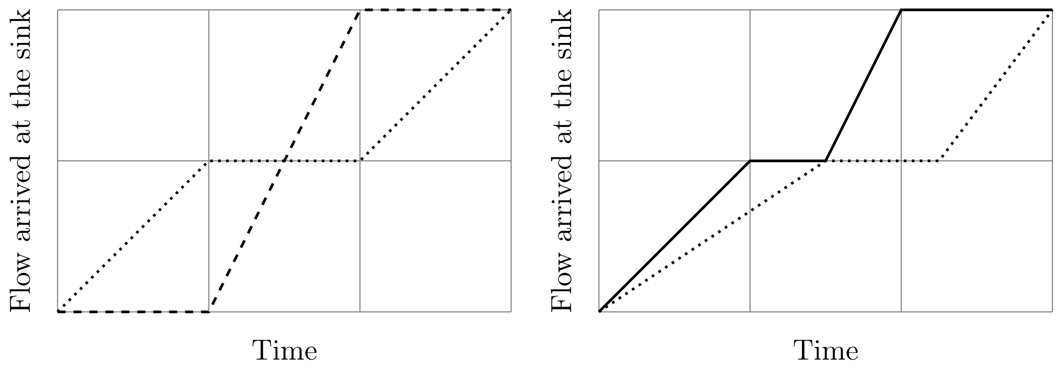


Figure 6.3: Network (V, A).

The structure of this instance is very similar to the instance in the previous theorem. Consequently, we have two options for fulfilling the supplies and demands. We either send the unit of commodity 1 first (which we will refer to as strategy A) or the unit of commodity 2 (which we will refer to as strategy B). Doing the latter will result in both units beginning to arrive by time 1 at the sink. Doing the former causes the first unit to begin arriving by time 0 and the second by time 2. An earliest arrival transshipment would need to have at least one unit of flow beginning to arrive by time 0 and an additional unit by time 1, which is not possible. Figure 6.4 (a) shows the amount of flow that has arrived at the sink by a certain time using the two strategies - the dotted line represents strategy A, the dashed line represents strategy B. Figure 6.4 (b) shows the maximal amount that can arrive at the sink by a certain amount of time, i.e. the amount of flow an earliest arrival transshipment would have to send by this time. The dotted line shows how much flow a $\frac{3}{2}$-earliest arrival transshipment would need to send by a specific time.

Strategy A sends 2 units of flow to the sink by time 3 whereas it is possible to send them to the sink by time 2. Strategy B sends 1 unit of flow to the sink by time $\frac{3}{2}$ whereas it is possible to send it to the sink by time 1. This shows that there is no strategy that yields an α-earliest arrival transshipment for $\alpha < \frac{3}{2}$ for this instance. It is easy to see that strategy A is a $\frac{3}{2}$-earliest arrival transshipment to this instance.



(a) Strategy A and B

(b) Earliest Arrival

Figure 6.4: Flow Arrival Patterns

□

If we consider instances with multiple sinks, we can construct even worse instances.

Theorem 6.2.3. *There are instances of the* Earliest Arrival Multicommodity Transshipment Problem with CDDT *and multiple sinks for which no α-earliest arrival transshipment exists, for $\alpha < M$ and some $M > 1$.*

Proof. Consider the following instance $I = (G = (V, A), K, b, \tau, u)$ of the Earliest Arrival Multicommodity Transshipment Problem with CDDT defined by $V := \{s_1, s_2, t_1, t_2\}$, $A := \{a = (s_1, t_1), b = (s_1, t_2), c = (s_2, t_2)\}$, $K := \{1\}$, $u_a = u_b = u_c = 1$, $b_{s_1} = b_{s_2} = 1$, $b_{t_1} = b_{t_2} = -1$ and $\tau_a = M$, $\tau_b = 1$, $\tau_c = M$. Figure 6.5 depicts the instance.

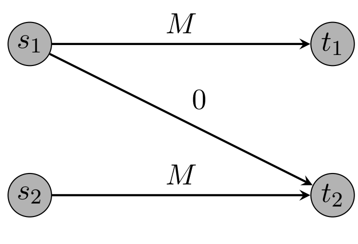


Figure 6.5: Network (V, A). The arc labels denote the transit time of the arcs.

It is possible to send one unit of flow along arc b causing it to begin arriving at t_2 by time 0. By time 1 one unit of flow will have arrived at the sinks. There is no way to send the second and last unit of flow from s_2 to t_1, though. Thus, any transshipment over time fulfilling all supplies and demands can only send flow along paths of length M. Therefore there are no α-earliest arrival transshipments for $\alpha < M$ for this instance. □

Note that the above result holds even if all sources can reach all sinks. In the above example we could add an arc $d = (s_2, t_1)$ with $u_d = 1$ and $\tau_d = M^2$ and would achieve the same result. Therefore we have to restrict our instances to those which have only a single sink for all commodities if we want to achieve better results. For similiar reasons we will assume from now on that no flow can arrive at sinks before time 1.

We will continue by developing a technique that allows us to construct α-earliest arrival transshipments for such instances. We assume for this sake that we are given an oracle $O(t)$ that for a given instance of the Earliest Arrival Multicommodity Transshipment Problem with CDDT provides us with a transshipment over time f that fulfills as many supplies and demands as possible for a given time t. We will now describe an algorithm which calculates a 4-earliest arrival transshipment for the Earliest Arrival Multicommodity Transshipment Problem with CDDT using O and an interval stacking technique as used in [2].

Let $I = (G = (V, A), K, b, \tau, u)$ be an instance of the Earliest Arrival Multicommodity Transshipment Problem with CDDT. We begin by using $O(t)$ to compute feasible transshipments over time f_j for $t = 2^j$, $j = 0, 1, \ldots$ iteratively until we find an f_k that fulfills all supplies and demands. Every $f_j, j < k$ fulfills the supplies and demands partially. It is our idea now to construct a transshipment over time f by sending the transshipments f_j, $j = 0, \ldots, k$ in succession, i.e. we send a transshipment over time f_j in the time interval $[2^j - 1, 2^{j+1} - 1)$. More formally, the flow value $f^i_a(\theta)$ at time θ for a commodity $i \in K$ and an arc $a \in A$ is equal to the flow value ${f_j}^i_a(\theta - (2^j - 1))$ if $\theta \in [2^j - 1, 2^{j+1} - 1)$. Figure 6.6 depicts some of the time intervals and their corresponding transshipments over time.

f_0	f_1	f_2	f_3

Figure 6.6: Time intervals for the $f_j, j = 0, 1, 2, 3$

The question now arises whether f defined as above is indeed a feasible transshipment over time that fulfills all supplies and demands. By construction, we know that f is feasible since at every time θ we are sending flow according to exactly one of the transshipments over time f_i, $i = 0, \ldots, k$; flow is only sent according to a transshipment over time f_j if all flow of transshipments over time $f_i, i < j$ has arrived at sinks. Thus, the capacity constraints hold. Due to f_i, $i = 0, \ldots, k$ being transshipments, we can also be sure that flow conservation holds for each individual f_i. If several f_i are sent in succession, however, it is possible that an f_j fulfills supplies and demands that have already been fulfilled by an $f_i, i < j$. Therefore, we need to "repair" the $f_i, i = 0, \dots, k$ iteratively to fulfill flow conservation when sent in succession. We make the following observation.

Observation 6.2.4. *Let I be an instance of a transshipment problem with a network $G = (V, A)$, supplies and demands b, capacities u, commodities K and transit times τ. Let f be a multicommodity transshipment over time fulfilling supplies and demands b. For any supplies and demands b' with $0 \le b'^i_v \le b^i_v$, $\forall i \in K$, $\forall v \in V : b^i_v > 0$ and $0 \ge b'^i_t \ge b^i_t$, $\forall i \in K$, $t \in V$ there is a multicommodity transshipment over time f' fulfilling supplies and demands b' with*

$$f'^i_a(\theta) \le f^i_a(\theta) \quad \forall a \in A, i \in K, \theta \in \mathbb{R}$$

Proof. Due to our assumption there is only a single node $t \in V$ with $b^i_t < 0$ for $\forall i \in K$. Since f fullfills the supplies and demands b it sends b^i_v units of flow of commodity i from v to t along some v-t-paths for all $v \in V : b^i_v > 0$. Obviously, we can reduce the amount of flow sent from v to t by sending less flow along these paths without violating capacity constraints. Flow conservation also only changes at the start node and the end node of the path, which is exactly what we are trying to accomplish. In order to fulfill the new supplies and demands, we need to reduce the value of flow travelling along v-t-paths by exactly $b^i_v - b'^i_v$ for every $i \in K$, $v \in V : b^i_v > 0$. The result is a transshipment over time f' fulfilling supplies and demands b' with

$$f'^i_a(\theta) \le f^i_a(\theta) \quad \forall a \in A, i \in K, \theta \in \mathbb{R}$$

□

Thus, we are able to obtain a sequence of transshipments over time $\overline{f}_i, i = 0, \dots, k$ that can be sent in succession by applying the above observation to $f_0, \dots, f_k$: if b' are the supplies and demands unfulfilled after $f_0, \dots, f_i$ have been sent, we modify f_{i+1} according to the observation so that sending f_{i+1} in the next time interval $[2^{i+1} - 1, 2^{i+2} - 1)$ does not violate flow conservation by sending more flow from a source than its supply allows. We can show this yields a 4-earliest arrival transshipment. In order to do this, we show the following lemma.

Lemma 6.2.5. *For a time θ let j be such that $\theta \in [2^j - 1, 2^{j+1} - 1)$. Then the following inequality holds:*

$$|f_{max}(2^{j-1})| = |f_{j-1}(2^{j-1})| \le \sum_{i=0}^{j-1} |\overline{f}_i(2^i)|$$

Proof. The first part, i.e,

$$|f_{max}(2^{j-1})| = |f_{j-1}(2^{j-1})|$$

follows directly from the definition of $|f_{max}(2^{j-1})|$ and $|f_{j-1}(2^{j-1})|$, respectively. We show the second part by induction over j. For $j = 0$, the inequality holds due to $f_0 = \overline{f}_0$ by construction. Now we assume that the inequality holds for all $0 \leq i < k$ and need to show that it holds for $j = k$ as well. Let B_{max}, $\overline{B}_{max}$ be the amount of supplies and demands fulfilled by f_k, $\overline{f}_k$, respectively, and let $\bar{b}_{kv}^{\,j}$ be the supply of commodity i fulfilled by $\overline{f}_j$ at node v. By construction of $\overline{f}_k$ we know that supply already fulfilled by a preceding transshipment might lower the amount of supply fulfilled by $\overline{f}_k$ itself, thus it follows that

$$|\overline{f}_k(2^k)| \geq |f_k(2^k)| - \sum_{i \in K} \sum_{v \in V : b_v^i > 0} \sum_{j=0}^{k-1} \bar{b}_{jv}^{\,i}$$

This is equal to

$$|\overline{f}_k(2^k)| \geq |f_k(2^k)| - \sum_{j=0}^{k-1} |\overline{f}_j(2^j)|$$

which is equivalent to

$$\sum_{j=0}^{k} |\overline{f}_j(2^j)| \geq |f_k(2^k)|$$

concluding the proof. □

Theorem 6.2.6. *The transshipment over time $\overline{f}$ as contructed above is a 4-earliest arrival transshipment.*

Proof. We have already seen that $\overline{f}$ is a feasible transshipment fulfilling supplies and demands.

For all $\theta \geq 0$ we define $\alpha_\theta \in \mathbb{R}$ as the smallest number, such that

$$|\overline{f}(\theta)| \geq |f_{max}\left(\frac{\theta}{\alpha_\theta}\right)|$$

holds. Thus $\max_{\theta \geq 0} \alpha_\theta$ is a lower bound for the best approximation of an earliest arrival transshipment that we can achieve, i.e. there are no α-earliest arrival transshipments with $\alpha < \max_{\theta \geq 0} \alpha_\theta$. In fact, if $\alpha = \max_{\theta \geq 0} \alpha_\theta$ for a feasible transshipment over time f fulfilling all supplies and demands then f is an α-earliest arrival flow by definition of α_θ. We will now show that $\alpha := \max_{\theta \geq 0} \alpha_\theta \leq 4$ holds for our transshipment over time $\overline{f}$.

For a time $\theta \geq 0$ let j be such that $\theta \in [2^j - 1, 2^{j+1} - 1)$. By the previous lemma, we know that

$$|\overline{f}(\theta)| \geq |f_{max}(2^{j-1})| \quad \forall \theta \in [2^j - 1, 2^{j+1} - 1)$$

If follows that

$$\alpha_\theta \leq \frac{\theta}{2^{j-1}} \leq \lim_{\epsilon \to 0} \frac{2^{j+1} - 1 - \epsilon}{2^{j-1}} \leq 4 \quad \forall \theta \geq 0$$

Thus, $\overline{f}$ is a 4-earliest arrival transshipment. □

We summarize our results so far. In the case of multiple commodities with commodity-dependent transit times, there are usually no earliest arrival transshipments nor α-earliest arrival transshipments with a guaranteed upper bound for α. If we confine ourselves to a single sink (and exclude instances where flow can reach the sink before time 1 for convenience) we can show that 4-earliest arrival transshipments exist for the Earliest Arrival Multicommodity Transshipment Problem with CDDT.

Computing such transshipments is unfortunately not easy, especially if the application requires Integer transshipments. The combination of commodity-dependent transit times and the Integer restrictions can lead to problems for which not even polynomial time approximation schemes exist as we have already seen in chapter 4.

Chapter 7

Conclusion

We conclude this thesis with a short summary and give possible topics for further research. In chapter 4 we introduced commodity-dependent transit times and discussed advantages and disadvantages of using them, especially the complexity of flow and transshipment problems in settings with commodity-dependent transit times. We came to the conclusion that commodity-dependent transit times increase the complexity of the problems even in cases that were easy to solve for commodity-independent transit times. Restricting the problems to integer flows - which is desirable for many applications - makes the problems even harder to solve or approximate. The violation of the first-in-first-out property is a natural consequence of having transit times that vary from commodity to commodity. The correct way to deal with this violation is rather application-specific, however, and therefore probably best explored in an application-specific context. It is quite possible that commodity-dependent transit times for a specific application contain enough structure to allow algorithms that are more efficient than what is possible in the general case. It would be an interesting topic of further research to examine commodity-dependent transit times in an application-specific context.

Based on the complexity results of chapter 4 we settled for developing a fully polynomial time approximation scheme for the multicommodity quickest transshipment problem with commodity-dependent transit times in chapter 5. This scheme is an extension of techniques developed by Fleischer and Skutella [7, 8] for the quickest flow over time problem, featuring condensed time-expanded networks as a key-component. Due to the complexity results of chapter 4, this result is best-possible from a theoretical point of view. It would be interesting to see if there are applications for which efficient algorithms for this problem exist.

In chapter 6 we considered multicommodity earliest arrival transshipment problems, especially the question of their existence and ways to approximate them. We saw that earliest arrival transshipments hardly ever exists in the context of commodity-dependent transit times. Even worse we cannot hope to find approximations in the form of α-earliest arrival transshipments if there are multiple sinks and commodity-dependent transit times. In the case of a single sink, we were however able to show that 4-earliest arrival transshipments exist.

List of Figures

3.1 Counterexample to the existence of earliest arrival flows. 23

4.1 Network (V, A_1). Edge labels denote transit times of the first commodity. . 33
4.2 Network $(V, A_i \cup \{(u_{i-1}, v_i)\}$ for $i \in \{2, \ldots, n+1\}$. Edge labels denote transit times of commodity i; the dashed arcs are only usable by commodity i and all arcs not shown are not usable by commodity i due to their respective transit times. The non-dashed arc is $(u_{i-1}, v_i) \in A_1$. 34
4.3 Network $(V, A_i \cup \{(w_{i-1}, v_i)\}$ for $i \in \{n+2, \ldots, 2n+1\}$. Edge labels denote transit times of commodity i; the dashed arcs are only usable by commodity i and all arcs not shown are not usable by commodity i due to their respective transit times. The non-dashed arc is $(w_{i-1}, v_i) \in A_1$. 34
4.4 Network (V, A_i), $i = 1, \ldots, n$. Edge labels denote transit times of commodity i. 37
4.5 Network (V, A_{n+1}) if clause c_1 has the form $c_1 = x_1 \vee \overline{x_2} \vee x_5$. Edge labels denote transit times of commodity $n+1$, edges not usable by commodity $i+1$ are not shown. 38

6.1 Network (V, A). 50
6.2 Network (V, A). 50
6.3 Network (V, A). 51
6.4 Flow Arrival Patterns 52
6.5 Network (V, A). The arc labels denote the transit time of the arcs. 52
6.6 Time intervals for the $f_j, j = 0, 1, 2, 3$ 53

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows. Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[2] N. Baumann and E. Koehler. Approximating earliest arrival flows with flow-dependent transit times. *Discrete Applied Mathematics*, 155:161–171, 2007.

[3] N. Baumann and M. Skutella. Solving evacuation problems efficiently: Earliest arrival flows with multiple sources. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 399–408, 2006.

[4] J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19:248–264, 1972.

[5] L. K. Fleischer. Faster algorithms for the quickest transshipment problem. *SIAM Journal on Optimization*, 12:18–35, 2001.

[6] L. K. Fleischer and M. Skutella. The quickest multicommodity flow problem. In *Lecture Notes in Computer Science 2337*, pages 36–53, 2002.

[7] L. K. Fleischer and M. Skutella. Minimum cost flows over time without intermediate storage. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 66–75, 2003.

[8] L. K. Fleischer and M. Skutella. Minimum cost flows over time without intermediate storage. *SIAM Journal on Computing*, 36(6):1600–1630, 2007.

[9] L. K. Fleischer and E. Tardos. Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23:71–80, 1998.

[10] L. R. Ford and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433, 1958.

[11] L. R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[12] D. Gale. Transient flows in networks. *Michigan Math. Journal*, pages 59–63, 1959.

[13] N. Garg and J. Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 300, Washington, DC, USA, 1998. IEEE Computer Society.

[14] A.V. Goldberg and R.E.Tarjan. A new approach to the maximum flow problem. *Journal of the ACM*, 35:921–940, 1988.

[15] M. Grötschel, L. Lovasz, and A. Schrijver. The ellipsoid algorithm and its consequences in combinatorial optimization. *Combinatorica*, 1:499–513, 1981.

[16] M. Grötschel, L. Lovasz, and A. Schrijver. *Network Flows. Geometric Algorithms and Combinatorial Optimization.* Springer Verlag, 1988.

[17] A. Hall, S. Hippler, and M. Skutella. Multicommodity flows over time : Efficient algorithms and complexity. *Mathematics of Operations Research*, 25:36–62, 2000.

[18] A. Hall, K. Langkau, and M. Skutella. An FPTAS for quickest multicommodity flows over time with inflow-dependent transit times. *Algorithmica*, 47(3):299–321, 2007.

[19] B. Hoppe and E. Tardos. Polynomial time algorithms for some evacuation problems. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 433–441, 1994.

[20] B. Hoppe and E. Tardos. The quickest transshipment problem. *Theoretical Computer Science*, 379(3):387–404, 2007.

[21] B. Klinz and G.J. Woeginger. Minimum cost dynamic flows: The series parallel case. *Networks*, 43:153–162, 2004.

[22] E. Köhler and M. Skutella. Flows over time with load-dependent transit times. *SIAM Journal on Optimization*, 15(4):1185–1202, 2005.

[23] B. Korte and J. Vygen. *Combinatorial Optimization.* Springer, 4. edition, 2008.

[24] E. Minieka. Maximal, lexicographic, and dynamic network flows. *Operations Research*, 21:517–527, 1973.

[25] D. Richardson and E. Tardos. *cited as personal communication in [8]*, 2002.

[26] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, 1986.

[27] A. Schrijver. *Combinatorial Optimization.* Springer, 2003.

[28] E. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5:247–255, 1985.

[29] W. C. Wilkinson. An algorithm for universal maximal dynamic flows in a network. *Operations Research*, 19:1602–1612, 1971.