2013-2

Tue Rauff Lind Christensen

PhD Thesis

# Network Design Problems with Piecewise Linear Cost Functions

DEPARTMENT OF ECONOMICS AND BUSINESS
AARHUS UNIVERSITY • DENMARK

# NETWORK DESIGN PROBLEMS
# WITH
# PIECEWISE LINEAR COST FUNCTIONS

by

Tue Rauff Lind Christensen

*A PhD thesis submitted to the School of Business and Social Sciences, Aarhus University, in partial fulfilment of the PhD degree in Economics and Business*

December 2012

Advisor: Kim Allan Andersen

# Contents

# Preface

In January 2009, I embarked on my journey towards a PhD degree surprisingly unprepared for life as a researcher. Looking back I realize just how far I have come and what an interesting experience it has been. On my way I was met with unforeseen challenges, but even though some of them seemed almost insuperable, I have now reached my destination through hard work (and a bit of luck). This PhD thesis is the academic fruit of the hard work and consists of four chapters, each of which constitutes an independent research paper. One of these papers has already been published (see Christensen et al. (2012)) and the rest are in the final stages before submitting them to journals. Overall I am very pleased with the academic outcome of my dissertation and feel I have contributed to the literature!

## Acknowledgment

A great number of people have helped and supported me when the challenges seemed too great. My supervisor Kim Allan Andersen infused me with confidence in my work through encouraging discussions. My unofficial co-supervisor Andreas Klose sparked my interest in Operations Research through several interesting courses and is now an inventive collaborator on several projects. My fellow PhD students supported me with cake, beer, and wonderful friendships. My colleagues at the *Cluster for Operations Research Applications in Logistics* always had time to discuss the problems at hand. My friends in Brussels provided a warmth and welcoming environment during my stay abroad, where Professor Martine Labbé made sure that the stay was challenging, but very rewarding. I am grateful toward the (former) Aarhus School of Business and Otto Mønsteds Fond for covering my expenses during my stay in Brussels.

I owe a great deal to my family for their never-ending love and support. To my friends both at the university and outside for the good company when I need to have a break. Finally, to my girlfriend, Sidse, for unconditional love and care!

*Tue Rauff Lind Christensen*
*October 2012*
*Aarhus N*

# Resumé

Denne ph.d.-afhandling betragter en række *netværksdesign problemer* og præsenterer metoder, der effektivt kan løse disse problemer. Netværksdesign problemer er, ofte lettere forsimplede, matematiske modeller for problemstillinger, der opstår i opbygningen af en virksomheds værdikæde og er en vigtig del af *supply chain management*. Det mest generelle problem, som undersøges i denne afhandling, kan beskrives som følger. Givet et antal mulige placeringer af fabrikker og et antal kunder med en kendt efterspørgsel, hvorledes skal fabrikkerne så placeres, således at omkostningerne ved at bygge fabrikkerne og servicere kunderne minimeres? Med en række antagelser vedrørende modelleringen af fabrikkerne og kunderne kan dette opstilles som et på en gang simpelt, men svært matematisk problem. Vi vil antage, at omkostningerne ved at bygge en fabrik er givet ved et engangsbeløb, mens transportomkostningerne ved at servicere en kunde følger en trappelignende omkostningsfunktion, også kendt som en stykvis lineær omkostningsfunktion. Denne afhandling bidrager med ny forskning specielt hvad angår det sidste punkt. Den trappe-lignende omkostningsstruktur er ikke blevet behandlet før, selvom den optræder i en række vigtige sammenhænge. Eksempler på den type omkostninger kan findes i fragt- og shippingsektoren, blandt andet i prisstrukturen for pakker ved det nationale postvæsen i en række lande inklusiv Danmark. Den trappe-lignende omkostningsstruktur udvider

den normale antagelse, at transportomkostningerne er konstante per sendt enhed. Målet med denne afhandling er at komme med forslag til mere effektive løsningsmetoder til disse matematiske problemer end dem vi hidtil har kendt.

Udover en introduktion og en litteraturgennemgang består afhandlingen af tre artikler. Den første artikel beskriver en meget effektiv løsningsmetode til det specialtilfælde, hvor der kun er en kunde og en række eksisterende fabrikker. Denne metode bruges i de resterende artikler til at løse en række subproblemer, der opstår undervejs i løsningen af de mere overordnede problemer. Den næste artikel ser på det mest generelle problem og præsenterer en række metoder til hurtigere løsning af den type problemer, når man benytter standard optimeringssoftware. Den sidste artikel beskæftiger sig udelukkende med transport aspektet i problemet. Givet et antal kunder og et antal lagre, hvorledes skal produkterne sendes til kunderne, når omkostningerne følger en stykvis lineær omkostningsfunktion? Vores forslag hertil er anvendelse af en specialiseret løsningsmetode baseret på en Dantzig-Wolfe reformulering af problemet.

# Abstract

This dissertation considers a number of *network design problems* and proposes solution methods to efficiently solve these problems. Network design problems are, often slightly simplified, mathematical models for problems arising in the construction of a company's supply chain and are an important part of *supply chain management*. The most general of the problems considered in this dissertation can be described as follows. Given a set of feasible facility locations and a set of customers with a known demand, how should the facilities be placed such that the total costs for building the facilities and servicing the customers are minimized? By making a number of assumptions concerning the facilities and customers, we can formulate this problem as an elegant, yet hard mathematical program. We assume that the costs for operating a facility at a site are a fixed amount and that the costs for servicing a customer follows a staircase-like structure. The latter assumption has not been treated previously within facility location and is a central part of the scientific contribution of this dissertation. This kind of cost structure is crucial when modeling certain important aspects of the supply chain and can be found in various parts of the freight and shipping industry, e.g. in the pricing of packages for numerous national postal services across the world, including Denmark. The staircase cost structure extends the normal assumption of a constant price per unit sent. The aim of this dissertation is to provide efficient solution methods for a

subset of these problems.

Besides an introduction and a literature review this dissertation consists of three main chapters/papers. The first of these main chapters proposes a very efficient solution method for a special case with only one customer and a set of open facilities. This solution method is used in the other two chapters as a subroutine in the solution methods for the more general problems. The second chapter treats the most general problem of the three papers and presents techniques to speed up the computation time required to solve the problem by state-of-the-art optimization software. The last chapter focuses on the pure transportation part of the problem. Given a set of customers and a set of open warehouses, how should items be shipped from the warehouses to the customers if the transportation costs follows a piecewise linear structure? For this problem we develop a specialized solution method based on a Dantzig-Wolfe reformulation of the problem.

# Introduction

In this thesis I will address three problems which share a number of commonalities. All of the problems are related to *Network Design* and all of them consider a *piecewise linear* cost structure. In this chapter I link the different projects together to provide a larger and more coherent picture with a common thread.

This chapter is outlined as follows. In the subsequent section I introduce the most general of the three problems considered, namely the *capacitated facility location problem with piecewise linear transportation costs* (CFLP with PLTC). Then we give an abstract of the remaining four chapters and relate the problems they address to each other.

## On the General Framework and Piecewise Linear Cost Functions

We consider a set of potential facility locations, denoted by $I = \{1, \ldots, n\}$ and a set of customers, denoted by $J = \{1, \ldots, m\}$. Each facility, if placed at site $i$, has a capacity limit of $S_i$ units, and the demand of customer $j$ is denoted by $d_j$. There is an associated fixed cost for opening a facility at site $i$ of $f_i$. Here we will consider a piecewise linear transportation cost function consisting of a set $Q_{ij} = \{1, \ldots, q_{ij}\}$ of segments (also known as modes) between the site $i$ and customer $j$. For notational convenience we will assume that $q_{ij} = q$ for all $(i, j)$. Such a cost function is motivated by the different ways of transporting goods, such as *small packages*, *less-than-truckloads* or *truckloads* (see Croxton et al. (2003b)), and is found in e.g. the shipping industry (Baumgartner et al. (2012)) and in the pricing of many national postal service's and couriers (Lapierre et al. (2004)). Alternatively, a piecewise linear cost function can be used to model price discounts, such as *all-unit* or *incremental* discounts that are often found in procurement theory (see Qi (2007) and Kameshwaran and Narahari (2009)). Each segment $l$ of the piecewise linear function between site $i$ and customer $j$ is characterized by four attributes (see Figure 1). First, the variable cost for using a mode (which, if applicable, also includes the per unit production cost of the facility). Secondly, the modes intercept with the $y$-axis, which can be interpreted as the fixed cost for using a mode. And finally, the minimum and maximum amount of goods that can be transported on the mode. These four characteristics are denoted by $c_{ijl}$, $g_{ijl}$, $L_{ij,l-1}$ and $L_{ijl}$, respectively. We will assume that $L_{ij0} = 0$. We define a variable $x_{ijl}$ as the flow on mode $l$ between site $i$ and customer $j$. Also,

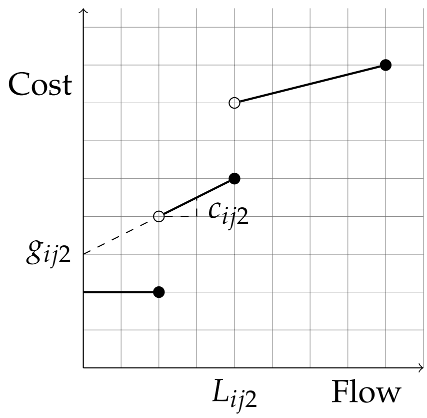



Figure 1: A piecewise linear cost function

we define the binary variable $v_{ijl}$ which is one if the aforementioned mode is used and zero otherwise. For each facility site $i$ we associate a binary variable $y_i$, which is one if a facility is located at the site and zero otherwise. The CFLP with PLTC can be formulated as

$$\text{(MCM)} \quad \min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{q} (c_{ijl} x_{ijl} + g_{ijl} v_{ijl}) + \sum_{i=1}^{n} f_i y_i \,, \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{l=1}^{q} x_{ijl} \geq d_j, \qquad \forall\, j, \tag{2}$$

$$\sum_{l=1}^{q} v_{ijl} \leq 1, \qquad \forall\, (i,j), \tag{3}$$

$$\sum_{j=1}^{m} \sum_{l=1}^{q} x_{ijl} \leq S_i y_i, \qquad \forall\, i, \tag{4}$$

$$x_{ijl} \leq L_{ijl} v_{ijl}, \qquad \forall\, (i,j,l), \tag{5}$$

$$x_{ijl} \geq L_{ij,l-1} v_{ijl}, \qquad \forall\, (i,j,l), \tag{6}$$

$$x_{ijl} \geq 0, \qquad \forall\, (i,j,l), \tag{7}$$

$$y_i \in \{0,1\}, \qquad \forall\, i, \tag{8}$$

$$v_{ijl} \in \{0,1\}, \qquad \forall\, (i,j,l). \tag{9}$$

The objective is to minimize the total costs consisting of the location and the transportation costs as stated in equation (1). The constraint (2) ensures that each customer's demand is met. This constraint could be modeled as an inequality if the cost function is non-decreasing (which seems reasonable in real-world applications). Equation (3) states that no more than one mode of transportation may be used between each pair of facility and customer. We are not allowed to exceed the maximum capacity at a facility by constraint (4). The constraints (5) and (6) ensure that the upper and lower bound on each mode are obeyed, respectively. The constraints (7), (8), and (9) define our decision variables.

## The Structure of the Thesis

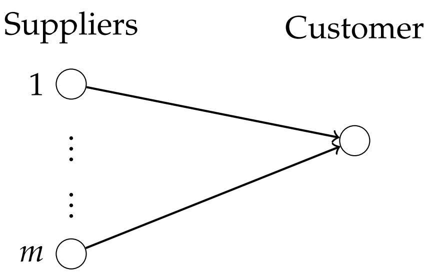
This thesis consists of four main chapters. In the following paragraphs we explain the contents of each chapter and relate these to the general framework presented in the previous section. The sequence of the chapters is identical to the time when the individual projects were started. Each chapter is an independent research paper and can be read as such. The only difference between the chapter and the corresponding paper is that the references have been gathered in a chapter at the end of the dissertation and minor journal specific details regarding layout have been leveled out.

**Chapter 1** is an overview of different facility location models that uses a non-standard production and/or transportation cost function such as the piecewise linear cost function. This was the first project of the thesis and was made to categorize existing literature. From this survey it was clear that the area of non-standard transportation costs had only been sparsely treated previously.

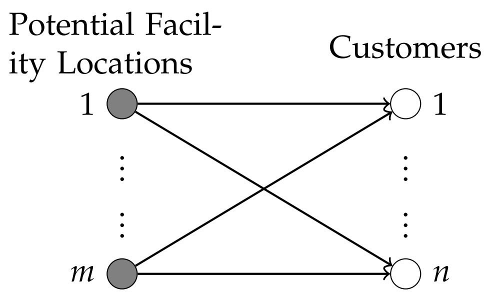
**Chapter 2** considers the *single-sink, fixed-charge, multiple-choice transportation problem* (SS-FCMCTP) and presents an efficient dynamic programming algorithm to solve it. The problem can be seen as a special case of the CFLP with PLTC where the fixed cost for opening facilities is zero, i.e. $f_i = 0$ for all $i$ (or the problem remaining when the facility sites have been fixed), and there is only one customer, i.e. $m = 1$ (see Figure 2a). The solution method extends several ideas known for the *single-sink, fixed-charged transportation problem*, such as *bound strengthening, variable pegging*, and *search space reduction* techniques. The paper has been accepted for publication in *Transportation Science* (see Christensen et al. (2012)).

**Chapter 3** treats the most general problem in this thesis, namely the CFLP with PLTC (see Figure 2b). We consider two formulations of the problem, namely the *multiple-choice model* (MCM) and the *discretized model* (DM) and investigate speed-up techniques for solving the problem using the standard mixed-integer programming solver CPLEX (version 12.4). The DM was proposed by Correia et al. (2010) for the capacitated facility location problem with *modular distribution costs* (where the terms distribution costs and transportation costs are used synonymously). As an aside we show how the problem with modular costs can be transformed into a problem with piecewise linear costs. We show that the two formulations have the same linear programming relaxation bound and propose a number of valid inequalities for the MCM similar to those proposed for the DM by Correia et al. (2010). Additionally, we propose a Lagrangean heuristic for the problem that obtains tight upper and lower bounds very fast and employs a variable pegging scheme to reduce the problem size. A number of computational tests show the impact of adding the valid inequalities and preprocessing the problem with the Lagrangean heuristic. These tests show that adding the valid inequalities strengthens the linear programming relaxation considerably. Using both the valid inequalities and the Lagrangean heuristic a total speed-up of up to 80% can be achieved compared to the plain model. The Lagrangean heuristic utilizes the dynamic programming algorithm from Chapter 2. The paper is not yet submitted, but the intended outlet is *Computers and Operations Research*.

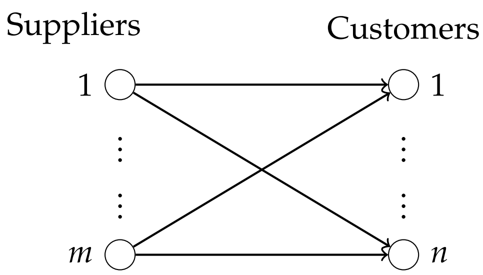
**Chapter 4** considers the *piecewise linear transportation problem* (PLTP), which is the special case of the CFLP with PLTC where the fixed cost of opening a facility is always zero, $f_i = 0$ for all $i$ (or, again, alternatively the facility sites have been fixed). Two new formulations of the problem are initially considered, both based on a Dantzig-Wolfe reformulation of the MCM. As both reformulations contain a very large number of variables we rely on *column generation* to solve the linear programming relaxation efficiently. The pricing problem is exactly the problem from Chapter 2. Computational tests show that one of the two formulations is much stronger than the other on the test instances used, and this model is then extended into an exact solution method. The latter is done by adding a number of valid inequalities based on violated *generalized upper bound constraints* and branching on the original variables if the linear programming relaxation is still fractional. The intended outlet is *Operations Research*.




(a) The problem considered in Chapter 2.




(b) The problem considered in Chapter 3. The most general of the problems considered.




(c) The problem considered in Chapter 4.

Figure 2: An Overview of the Problems Treated in this Dissertation.

# Chapter 1

# A Literature Review on Discrete Facility Location with Non-Standard Cost Functions

# A Literature Review on Discrete Facility Location with Non-Standard Cost Functions

Tue R. L. Christensen and Kim Allan Andersen

Department of Economics and Business, Aarhus University, Denmark

{tuec, kia}@asb.dk

**Abstract**

In this paper we review the literature on the discrete facility location with non-standard cost functions. Such models often arise when considering real world extensions such as waiting time penalization, inventory holding cost and economics of scale. We focus on the solution methods proposed and divide the literature into three main streams by categorizing the cost functions used. The paper is concluded with some remarks.

## 1.1 Introduction and Basic Model

The discrete facility location problem is a well-studied problem within operational research and many different extensions and applications of the problem have emerged over the years. Previous reviews concerning the standard facility location problem and its solution can be found in Sridharan (1995) and Hamacher and Nickel (1998). Specialized overviews concentrating on the facility location problem's role in supply chain management can be found in Klose and Drexl (2005a) and Melo et al. (2008). ReVelle et al. (2008) give an overview of different areas in location science such as median, center and cover problems as well as discrete facility location. The literature on non-standard cost functions is only sporadically mentioned in these reviews, however, many ideas are transferable from one problem to another. The aim of this article is to list these articles and highlight the solution method used, ideally providing valuable insight when faced with similar problems.

Literature on facility location models in the presence of inventory holding costs is abundant. The problems are fundamental within *integrated supply chain design* for which a survey is presented in Shen (2007). One approach is to approximate the safety stock costs and incorporate them into the fixed cost of a facility, as done in Nozick and Turnquist (1998, 2001). Other approaches assume the simple Economic Order Quantity (EOQ) model, which leads to a nonlinear term which can be efficiently solved under certain assumptions when using either a Lagrangean heuristic (as in Daskin et al. (2002)) or by solving the problem exactly using a set covering formulation (as in Shen et al. (2003)). In this paper we will not review the literature dealing with this kind of models.

In the last part of this section we formally introduce the *capacitated facility location problem*. The overview is split into three main sections. Section 1.2 deals with the case of concave cost functions and in particular the area of location-inventory models, where such a cost structure arises. Section 1.3 looks at convex cost functions arising in e.g. a service system design problem. In Section 1.4 we review the literature on piecewise linear cost functions. Finally, in Section 1.5 we review cost functions not falling into any of the aforementioned categories. We conclude this paper with an outline of trends and some observations in Section 1.6.

### 1.1.1 The Facility Location Problem

The facility location problem deals with deciding where to open facilities and how to assign customers to these facilities. The possible locations of the facilities and the customer locations are given. The placement of the facilities should be done such that the total costs are minimized. The total costs are comprised of three components: The fixed cost of opening and operating a facility, the cost of throughput (or production) and the cost of transporting goods to the customers. In this standard model the latter two are usually added into one term. Let $J = \{1, \dots, m\}$ denote the set of customers and $I = \{1, \dots, n\}$ the set of possible facility locations. Let $d_j$ denote customer $j$'s demand, and let $c_{ij}$ be the unit cost of serving customer $j$ from site $i$. Let the fixed cost (setup cost) for opening a facility at site $i$ be $f_i$. At each potential facility site $i$ there is an associated capacity of $S_i$. There are two kinds of decision variables, namely $x_{ij}$ and $y_i$. $x_{ij}$ is a continuous variable denoting the flow between facility $i$ and customer $j$. $y_i$ is a binary variable which is one if a facility is placed at site $i$ and zero otherwise. The capacitated facility location problem (CFLP) can be stated as follows

$$
\begin{aligned}
\text{(CFLP)} \quad \min \quad & \sum_{i=1}^{n} y_i f_i + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} && && (1.1)\\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = d_j && j = 1, \dots, m && (1.2)\\
& \sum_{j=1}^{m} x_{ij} \le y_i S_i && i = 1, \dots, n && (1.3)\\
& x_{ij} \ge 0 && i = 1, \dots, n, \ j = 1, \dots, m && (1.4)\\
& y_i \in \{0, 1\} && i = 1, \dots, n && (1.5)
\end{aligned}
$$

Equation (1.1) states thatwe are to minimize the total costs. The demand constraint, (1.2), ensures that all customers are fully served. Constraint (1.3) ensures that a customer cannot be served from a site where no facility is located and imposes the capacity restriction on an open facility. The constraints (1.4) and (1.5) defines our decision variables. In some cases the capacity on each facility might be unlimited or irrelevant for the decision process in which case the problem is known as the *uncapacitated facility location problem* (UFLP). The latter problem can be stated by replacing $S_i$ by a sufficiently large constant (e.g. $\sum_{j=1}^{m} d_j$) or by scaling the demand to one and then simply drop $S_i$ from the constraint (1.3). Both the UFLP and the CFLP are NP-hard as shown in Mirchandani and Jagannathan (1989). This literature review covers articles that conceptually extend any of the three cost components (fixed cost, transportation cost and production cost). In general, however, we refrain from reviewing multi-period, multi-stage, multi-product *etc.* models when they essentially reduces to a CFLP/UFLP when dropping the "multi"-aspect of the problem.

## 1.2 Concave Cost Functions

The per unit cost of delivering a product to a set of customers from a warehouse is often decreasing with the volume delivered. Thus, the cost may be represented by a concave function. Therefore, a common reason for looking at concave functions stems from economies of scale arising as a result of e.g. quantity discounts, specialization, increasing returns to scale, *etc*.

In this section we will replace the objective function (1.1) with the following objective function:

$$\text{(Concave)} \quad \min \quad \sum_{i=1}^{n} y_i f_i + \sum_{i=1}^{n} p_i(\sum_{j=1}^{m} x_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ij}(x_{ij}), \tag{1.6}$$

where $p_i(\cdot)$ is the concave production cost function at facility $i$ and $t_{ij}(\cdot)$ is the concave transportation cost function between facility $i$ and customer $j$. In some of the papers reviewed the setup costs are not considered explicitly but are merely part of the concave production costs.

**Feldman et al. (1966)** consider an UFLP with no setup costs where the production cost function is continuous and concave and the transportation costs are linear. They extend the ideas of Kuehn and Hamburger (1963) and the well-known "DROP" and "ADD" heuristics to the continuous, concave case. The scheme is iterative and in each iteration only a subset of the potential facility sites is considered. In the experimental results they restrict themselves to a piecewise linear continuous concave function with either one or two segments.

In **Zangwill (1968)** the main research question concerns flows in networks, but an UFLP with no setup costs is also considered. The production costs, as well as the transportation costs, are assumed to be concave and the problem is formulated as a *single commodity, acyclic, single source, multiple destination problem*. A dynamic programming algorithm is presented to solve the problem, but no experimental results are given.

**Soland (1974)** presents a branch and bound algorithm for general concave production and transportation costs for the CFLP. Soland modifies a more general method of minimizing separable concave functions over a linear polyhedron. The method approximates the concave functions by underestimating them with piecewise linear functions and solving a series of transportation problems. Branching amounts to replacing a segment of the approximation by two new segments, which improves the approximation and ensures finiteness of the solution method.

In **Kelly and Khumawala (1982)** a CFLP with setup costs, continuous, concave production costs and linear transportation is considered. They iteratively solve transportation problems resulting from approximating the cost function with either a tangent or chord approximation. Exploiting two propositions they fix the facility sites to be open or closed and eventually end up with the optimal solution.

**Kubo and Kasugai (1991)** consider a CFLP with no setup costs, concave production costs and concave transportation costs. The authors present a Lagrangean heuristic by making a

Lagrangean relaxation of the demand constraint (1.2) and linearize the cost functions. Feasible solutions are found by opening desirable facility sites (based on the solution of the Lagrangean subproblem) and solving the resulting transportation problem.

**Dasci and Verter (2001)** consider a facility location problem with multiple products, no capacity restriction, setup costs and linear transportation costs. The production and technology acquisition costs follows a continuous concave function as a function of the total amount of each product type that is served from a given facility site. This cost is underestimated using a piecewise linear function and the resulting problem is thus an UFLP with piecewise linear production costs (and therefore the paper could be placed in Section 1.4). The problem is solved by branch-and-bound and the lower bound of each node is calculated by a specialized dual-ascent method. The branching ensures that the linearization is gradually improved until the optimal solution is found.

**Hajiaghayi et al. (2003)** consider an UFLP, in which the production cost function of a facility is modeled as a concave function of the number of customers assigned to it, but with zero setup cost. This is motivated by an application of locating internet servers. They develop a greedy heuristic and proves that it has an approximation ratio of 1.861 for metric transportation cost. For the non-metric case the approximation ratio is $\ln(m+n)$. For the case of a convex production cost function they prove that the problem can be solved in pseudo-polynomial time.

In **Dupont (2008)** it is assumed that the production costs, transportation costs and setup costs all are concave functions. The setup cost is a function of the size of the facility (doubling the size of a facility does not double the setup cost). In this UFLP setting the author proves two properties of an optimal solution. One of them is used to characterize dominated solutions, from which a branching rule is derived. In the experimental results each facility is restricted to serve only customers within a certain radius and the effect on the performance of this is investigated.

## 1.3 Convex Cost Functions

Often, it is possible to expand production by incurring some additional costs. These costs may arise from overtime wages, accelerated depreciation or the use of more expensive materials from a distant supplier and are often modeled as a convex function.

In this section we will replace the objective function (1.1) with the following objective function:

$$\text{(Convex)} \quad \min \quad \sum_{i=1}^{n} y_i f_i + \sum_{i=1}^{n} p_i(\sum_{j=1}^{m} x_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ij}(x_{ij}), \tag{1.7}$$

where $p_i(\cdot)$ is the convex production cost function at facility $i$ and $t_{ij}(\cdot)$ is the convex transportation cost function between facility $i$ and customer $j$.

In **Mirchandani and Jagannathan (1989)** the total setup costs are given as a function of the number of open facilities for a UFLP. The authors start by formulating the problem for a general setup cost function, but later assumes a convex function and restricts the solution to be integral. For this special case they propose an algorithm relying on the convex function and presents a number of propositions on the optimal number of open facilities. A bisection search for the optimal number of facilities is then conducted guided by the DUALOC procedure (by Bilde and Krarup (1977) and Erlenkotter (1978)) and the solutions to fixed costs $p-$median problems.

In **Holmberg (1999)** an UFLP with setup costs and convex transportation costs is presented. Restricting the flow variables to be integral the problem is then reformulated by introducing binary variables on each arc, for each feasible flow. To solve this problem the authors embed a dual ascent method in a branch and bound framework. They also propose a Bender's decomposition for the problem. Computational results did not clearly favor one of the proposed methods.

The problem considered in **Harkness and ReVelle (2003)** is an UFLP with setup costs and convex, piecewise linear, increasing production costs. They give three formulations for the general setting with an arbitrary number of segments and one for the special case with only two line segments (their formulation for the special case is, however, different from the one proposed by Efroymson and Ray (1966)). They also propose otherwise redundant constraints to help improve the LP relaxation value. In order to test the different formulations they solve them to optimality using a standard MIP solver. On the randomly generated test cases one formulation seems superior with respect to the solution time. The framework of this paper is to be seen as a special case of the general piecewise production cost functions, treated in Section 1.4.

## 1.4 Piecewise Linear Cost Functions

In this section we review the literature on piecewise linear cost functions. When such a function is non-decreasing it is also known as a *staircase cost function*. Several realistic problems can be modeled using this kind of cost function. One example is the case where there is a setup cost associated with production, the production cost per unit is constant within a limited number of intervals described by threshold values, and the production cost per unit decreases whenever production exceeds a threshold value. Another example is the case when the decision to build a facility at a given site also requires a decision on the amount of capacity to be installed at the facility. Similarly there are examples where the transportation costs are modeled as a piecewise linear function, for instance when the unit transportation price decreases when larger volumes are transported.

In this section we will replace the objective function (1.1) with the following objective function:

$$\text{(Piecewise Linear)} \quad \min \quad \sum_{i=1}^{n} p_i(\sum_{j=1}^{m} x_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ij}x_{ij}, \tag{1.8}$$

where $p_i(\cdot)$ is the piecewise linear production cost function at facility $i$ and $t_{ij}$ is the (linear) transportation cost per unit between facility $i$ and customer $j$. Notice, that the fixed costs are handled implicitly by the piecewise linear production cost function.

If we assume that the piecewise linear production cost function at facility $i$ consists of $K$ line segments it may be expressed in this way:

$$\sum_{i=1}^{n} p_i(\sum_{j=1}^{m} x_{ij}) \quad = \quad \sum_{i=1}^{n}\sum_{k=1}^{K} f_{ik}y_{ik} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{K} p_{ijk}x_{ijk}, \tag{1.9}$$

where $y_{ik}$ is a binary variable which is one if we are operating at production level $x_{ijk}$ corresponding to line segment $k$. The corresponding fixed and variable costs are $f_{ik}$ and $c_{ijk}$, respectively.

The main contribution in **Efroymson and Ray (1966)** is a branch-and-bound algorithm for the UFLP for which they also discuss several extensions. One of these extensions is a piecewise linear, continuous, concave production cost function. They give a reformulation for the special case with exactly two line segments and discuss how to extend it to an arbitrary number of segments.

In **Holmberg (1994)** a number of solution methods for the CFLP with linear transportation costs and a staircase cost production function are proposed. This includes a Bender's decomposition, a cross decomposition, and a number of branch and bound methods. The staircase cost function is approximated by its convex envelope and this approximation is then improved from one iteration to the next in the solution method. Computationally, the results are inconclusive since every solution method is best in some setting. The same problem is considered in **Holmberg and Ling (1997)**. Here several variants of a Lagrangean heuristic are proposed. The relaxed constraints are the coupling between the production at a facility and the amount sent from this facility. Additionally, they modify the "ADD" heuristic in order to serve as a benchmark for their own heuristic. The computational results indicate that their Lagrangean heuristic produces high quality solutions in a reasonable amount of time.

The problem under consideration in **Harkness and ReVelle (2002)** is the CFLP with staircase production cost. To solve this the authors propose a more sophisticated Lagrangean heuristic than Holmberg and Ling (1997) by also employing variable pegging to fix the facility location variables. They perform an analysis of the performance of their heuristic. The results suggests that the most important factor for a good performance is the ratio between the manufacturing and transportation costs.

**Correia and Captivo (2003)** also considers the CFLP with linear transportation costs and piecewise linear production costs, but with a small extension, since they allow for certain production ranges to be inaccessible (i.e. holes in the staircase). Independently of Harkness and ReVelle (2002), they develop a similar Lagrangean heuristic. In addition, they calculate bounds on the maximum and minimum number of facilities that should be open, which is exploited by the heuristic. In **Correia and Captivo (2006)** the Lagrangean heuristic from Correia and Captivo (2003) is extended to the case where single sourcing is required, that is, every customer must be served by exactly one facility. The heuristic is extended with a tabu search or a local search procedure in order to improve the solution.

Motivated by a restructuring process, **Broek et al. (2006)** considers the placement of Norwegian slaughterhouses. Data supplied by slaughterhouses indicates that the production cost function is a piecewise linear function, but has a strong resemblance to a convex function. The resulting problem is a CFLP with linear transportation costs and a staircase production cost function. They develop a Lagrangean heuristic, since earlier work showed the problem is poorly solved by a simple branch and bound scheme (in Borgen et al. (2000)). The Lagrangean heuristic developed is similar to Harkness and ReVelle (2002) and Correia and Captivo (2003), but developed independently.

**Wollenweber (2008)** uses a staircase cost function to model the production costs at a car dismantling facility. The transportation costs are linear. The overall problem is modeled as a multistage facility location problem with three stages: recollection, dismantling, and recycling. This multi-stage problem is solved heuristically by combining ideas from greedy algorithms, “ADD”, “DROP”, exchange, and variables neighborhood search heuristics.

The slaughterhouse case study of Broek et al. (2006) is also the motivation in **Schütz et al. (2008)**. The model consists of two stages. The first stage is a long run decision, where the location and size (a range) are chosen and the cost function used is a piecewise linearization of an S-shaped cost function. In the short run the costs are convex and within the range of the long run cost function, agrees with it (outside the cost is higher). Hence, one strives to choose the range (long run decision) to match the actual demand assigned to the facility in the short run. To solve this a Lagrangean heuristic is employed by relaxing the demand constraint of the second stage of the problem (the assignment of customers for given capacities at the facilities).

One observation to be made in this section is that most of the papers use Lagrangean relaxation, relaxing the demand constraints, as part of the solution procedures. In general this works very well.

## 1.5 Other Non-Linear Cost Functions

There are many reasons why a cost function may be neither concave, convex or piecewise linear. In particular this may be the case when one initially has economies of scale and then reaches a point at which production is stretched into a less efficient range, as in the S-shaped function suggested by microeconomic production theory (see e.g. Henderson and Quandt (1971)).

In this section we will replace the objective function (1.1) with the following objective function:

$$\text{(Other)} \quad \min \quad \sum_{i=1}^{n} f_i(y_i) + \sum_{i=1}^{n} p_i(\sum_{j=1}^{m} x_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ij}(x_{ij}), \tag{1.10}$$

where $f_i(\cdot)$ is the setup cost function at location $i$, $p_i(\cdot)$ is the production cost function at facility $i$ and $t_{ij}(\cdot)$ is the transportation cost function between facility $i$ and customer $j$.

The main contribution in **ReVelle and LaPorte (1996)** is the description of several extensions of the UFLP and CFLP, which all have practical relevance. Among those are the maximum return-on-investment plant location problem (a linear fractional objective function is proposed), a biobjective plant location problem, and plant location problems with spatial interaction. The last one may give rise to a convex objective function, see Holmberg (1999). Another interesting problem proposed is the *single product, capacitated machine siting problem*. Besides locating facilities (setup costs) and assigning customers to these (transportation costs), the problem consist of choosing a number of machines to be placed at each facility. The number of installed machines determines the capacity at the facility and a fixed cost for each machine is incurred.

The service system design problem is basically the capacitated facility location problem with stochastic demand and a waiting time penalty. The goal is to locate service facilities and allocate customers to these so as to minimize total costs. The cost function comprises service (production) costs, fixed cost of placing a facility at a site, and a waiting term penalty if a customer cannot be served immediately. This problem is considered in **Amiri (1997)** who introduces a non-linear integer programming formulation and proposes two Lagrangean heuristics for the problem. The constraint relaxed is the equality constraint resulting from variable splitting (see e.g. Guignard and Kim (1987)). The difference between the two heuristics is the effort put into making a feasible solution in each iteration of the subgradient procedure. The paper presents experimental data that indicates that the additional effort put into the computationally heavier heuristic is well spent. In terms of solution quality it is considerably better.

The problem considered in **Elhedhli (2006)** is similar to the one considered in Amiri (1997). The solution method proposed can produce solutions arbitrarily close to the optimal solution. The cost function is then approximated by a piecewise linear function defined by a set of tangents to the original function. This approximation is then iteratively improved by cutting planes until the optimal solution to the original problem is found.

**Averbakh et al. (1998)** presents an UFLP with transportation costs and where the setup cost is dependent on the number of customers assigned to it. In particular, the authors consider the special case where the underlying network is a tree and for this they develop a polynomial time dynamic programming algorithm. It first considers the leaves of the tree and then successively moves toward the root. The optimal solution is then found by backtracking. No computational results are given. The work has been extended in **Averbakh et al. (2007)**. The customers choose which facility should serve them based on a price charged by the facility. The price is a function of the number of customers connected to the facility as well as the location. Each customer also has to account for the fact that a fraction of the transportation cost should be paid by him (the rest by the designer). The system designer has to place facilities such that the profit is maximized or the system's costs are minimized.

When the customers and facilities are located on a tree, a dynamic programming algorithm is proposed with similarities to that of Averbakh et al. (1998).

**Holmberg and Tuy (1999)** presents a CFLP with stochastic demand and concave production costs. They model the stochastic nature of demand by introducing a convex penalty function representing the shortage and holding costs at the customers. The overall objective function is the sum of linear transportation costs, concave production costs and convex shortage/holding costs. Thus, it can be seen as the difference between two convex functions (a d.c. function). The solution method is based on the fact that for fixed sizes of the facilities the assignment of customers to these facilities can be done efficiently by solving a *stochastic transportation problem*. The concave production cost is linearized and branching amounts to improve this approximation by splitting one segment into two.

In **Cañavate-Bernal et al. (2000)** several solution methods are presented, all based on Lagrangean relaxations for the single product, capacitated machine siting problem. This is one of the extensions presented in ReVelle and LaPorte (1996). In total three Lagrangean relaxations are proposed. Two of them relax the demand constraint and the aggregated demand constraint, respectively, while the last relaxation is based on variable splitting. In tests only the relaxations based on variable splitting and relaxation of the demand constraint seems to be competitive.

**Wu et al. (2006)** considers a CFLP where each customer demands several types of product. To serve these customers facilities of different types have to be placed and multiple facilities may be located at each site. There is an opening cost for each site and also a general setup cost depending on the number of facilities of a certain type placed at the site. Transportation costs are also taken into account. Two different formulations are presented and their relative strengths are tested by branch and bound with a standard solver (similarly to

Harkness and ReVelle (2003)). Results are inconclusive as one formulation is the better in case of concave setup costs and the other is better in case of convex setup costs. As a more practical solution method they propose a Lagrangean heuristic based on the relaxation of the demand constraint.

In **Gabor and van Ommeren (2006)** the authors define an UFLP with a subadditive total cost (setup and transportation) function. They show a reduction to the CFLP with soft capacity constraints for which a $2(1+\epsilon)-$approximation is known. For a special case of subadditive costs they propose a $2-$approximation algorithm based on a reduction to a facility location problem where the total cost function is linear.

**Correia et al. (2010)** considers a CFLP with modular distribution costs. Besides standard transportation costs, one has to decide, from a set of modules, each with a given fixed cost and capacity, how many to place on each arc to ensure sufficient capacity. Note, that this is essentially the equivalent arc cost function of the machine siting problem considered by ReVelle and LaPorte (1996) and Cañavate-Bernal et al. (2000). The authors formulate a basic model and a stronger discretized model and propose several valid inequalities for both models. Through computational tests they show that the discretized model with valid inequalities is superior to the basic model.

## 1.6 Concluding Remarks

Although non-standard costs arise in a number of different applications trends are observable and we will highlight them in this section and we will provide a classification of the papers reviewed, see Table 1.1.

One general observation is that almost all papers suggest a mixed-integer programming formulation, and only a few papers consider exact solution methods. The cost components in the objective function are the setup costs, transportation costs and production costs. In general the non-linear part is the setup costs and the production costs. Most of the reviewed papers contain experimental results.

We have reviewed thirteen papers which solve UFLP with a non-linear objective function. In most papers the cost components in the objective function were concave or convex. None of the components in the objective are piecewise-linear. The most popular exact solution methods are Branch-and-Bound and dynamic programming. A few papers also suggest heuristics.

Eighteen papers were concerned with CFLP with a non-linear objective function. In most papers the cost components in the objective function were concave or piecewise linear. Almost all solution methods are heuristics and most papers suggest the use of Lagrangean relaxation with relaxed demand constraints. There is no doubt that the use of Lagrangean relaxation has been a popular and fruitful solution method.

Table 1.1: Classification of the reviewed papers.

| Paper | Problem type | Cost components | Objective function | Solution method | Implemented |
|---|---|---|---|---|---|
| **Feldman et al. (1966)** | UFLP | Production | Concave | Heuristic | Yes |
| **Zangwill (1968)** | UFLP | Production & Transportation | Concave | Dynamic Programming | No |
| **Soland (1974)** | CFLP | Production & Transportation | Concave | Branch and Bound | Yes |
| **Kelly and Khumawala (1982)** | CFLP | Production & Transportation | Concave | Exact iterative procedure | No |
| **Kubo and Kasugai (1991)** | CFLP | Production & Transportation | Concave | Lagrangean relaxation | Yes |
| **Dasci and Verter (2001)** | UFLP | Production | Concave | B&B | Yes |
| **Hajiaghayi et al. (2003)** | UFLP | Production | Concave or convex | Heuristic | No |
| **Dupont (2008)** | UFLP | Production & Transportation | Concave | Branch and Bound | Yes |
| **Mirchandani and Jagannathan (1989)** | UFLP | Production | Convex | Heuristic | |
| **Holmberg (1999)** | UFLP | Transportation | Convex | Branch and Bound, Benders decomposition | Yes |
| **Harkness and ReVelle (2003)** | UFLP | Production & Transportation | Convex | Branch and Bound | Yes |
| **Efroymson and Ray (1966)** | UFLP | Production | Staircase | Branch and Bound | No |
| **Holmberg (1994)** | CFLP | Production & Transportation | Staircase | B&B, Benders and cross decomposition | Yes |
| **Holmberg and Ling (1997)** | CFLP | Production | Staircase | Lagrangean relaxation | Yes |
| **Harkness and ReVelle (2002)** | CFLP | Production | Staircase | Lagrangean relaxation | Yes |
| **Correia and Captivo (2003)** | CFLP | Production & Transportation | Staircase | Lagrangean relaxation | Yes |
| **Correia and Captivo (2006)** | CFLP | Production & Transportation | Staircase | Lagrangean relaxation | Yes |
| **Broek et al. (2006)** | CFLP | Production & Transportation | Staircase | Lagrangean relaxation | Yes |
| **Wollenweber (2008)** | CFLP | Production & Transportation | Staircase | Heuristic | Yes |
| **Schütz et al. (2008)** | CFLP | Production & Transortation | Non-linear | Lagrangean relaxation | Yes |
| **ReVelle and LaPorte (1996)** | UFLP/CFLP | Production & Transportation | Staircase | None | No |
| **Amiri (1997)** | CFLP | Transportation | Non-linear | Heuristic | Yes |
| **Elhedhli (2006)** | CFLP | Transportation | Non-linear | Cutting planes | Yes |
| **Averbakh et al. (1998)** | UFLP | Transportation | Non-linear | Dynamic programming | No |
| **Averbakh et al. (2007)** | UFLP | Transportation | Non-linear | Dynamic programming | No |
| **Holmberg and Tuy (1999)** | CFLP | Production & Transportation & Shortage | d.c. | B&B | Yes |
| **Cañavate-Bernal et al. (2000)** | CFLP | Transportation | Staircase | Lagrangean relaxation | Yes |
| **Wu et al. (2006)** | CFLP | Transportation | Non-linear | Lagrangean relaxation | Yes |
| **Gabor and van Ommeren (2006)** | UFLP | Transportation | Subadditive | Heuristic | No |
| **Correia et al. (2010)** | CFLP | Transportation | Non-linear | CPLEX + valid inequalities | Yes |

# Chapter 2

## Solving the Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem by Dynamic Programming

# Solving the Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem by Dynamic Programming

Tue R. L. Christensen†, Andreas Klose* and Kim Allan Andersen†

†Department of Economics and Business, Aarhus University, Denmark

{tuec, kia}@asb.dk

*Department of Mathematics, Aarhus University, Denmark, aklose@imf.au.dk

---

**Abstract**

This paper considers a minimum-cost network flow problem in a bipartite graph with a single sink. The transportation costs exhibit a staircase cost structure because such types of transportation cost functions are often found in practice. We present a dynamic programming algorithm for solving this so-called *Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem* exactly. The method exploits heuristics and lower bounds in order to peg binary variables, improve bounds on flow variables and reduce the state space variable. In this way, the dynamic programming method is able to solve large instances with up to 10,000 nodes and 10 different transportation modes in a few seconds, much less time than required by a widely-used MIP solver and other methods proposed in the literature for this problem.

---

*This paper has been accepted for publication in Transportation Science. Reprinted by permission, Christensen, T. R. L., K. A. Andersen, A. Klose. 2012. Solving the single-sink, fixed-charge, multiple-*

*choice transportation problem by dynamic programming. Transportation Science. Copyright 2012, the Institute for Operations Research and the Management Sciences, 7240 Parkway Drive, Suite 300, Hanover, Maryland 21076 USA.*

## 2.1 Introduction

The *Single-Sink, Fixed-Charge Transportation Problem* (SSFCTP) is to find a network flow from a set of suppliers with given supplies to a single sink that meets this sink's demand at minimum total cost. The shipping cost comprises a fixed charge and a linear term proportional to the amount of flow. In many cases the assumption of such a relatively simple transportation cost structure is, however, too limiting. First, the transportation costs faced by many industries are in fact piecewise linear. This can be seen in the actual freight rates for chemical companies (Baumgartner et al., 2012), in generic distribution networks (Croxton, Gendron, and Magnanti, 2003a; Lapierre, Ruiz, and Soriano, 2004; Croxton, Gendron, and Magnanti, 2007) and in the special case of merge-in-transit distribution networks (Croxton, Gendron, and Magnanti, 2003b). Second, the SSFCTP does not take price discounts into account, albeit such discounts are typically encountered within procurement and auctions (see, e.g., Davenport and Kalagnanam (2001) and Kameshwaran and Narahari (2009)).

Introducing staircase transportation cost functions in the SSFCTP leads to an extended problem formulation with multiple choices regarding the transportation modes to be applied to each link. We call this problem the *Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem* (SSFCMCTP) and it is, by extending the SSFCTP, $\mathcal{NP}$-hard. The SSFCTP is already a difficult problem (see, e.g., Klose (2008)), and by allowing for a staircase cost structure an even harder problem arises due to the additional binary variables required to model this cost function. Kameshwaran and Narahari (2009) consider the same optimization problem and call it a nonconvex piecewise linear knapsack problem for which they propose a dynamic programming algorithm. However, our computational results will show that their approach is strongly outperformed by the method presented in this paper. The model is actually a network flow problem with (nonconvex) piecewise linear cost in a simple network connecting a single sink with a set of customers. This model is versatile and allows for modeling cost accurately in important application areas such as supplier selection with price discounts and transportation costs when faced with alternative modes, such as *small packages*, *less-than-truckloads*, *full truckloads* and *air shipments*. The SSFCMCTP moreover arises as a relaxation of more general minimum-cost network flow problems with piecewise linear costs and of discrete facility location problems that show such types of transportation cost functions.

In this paper, we propose a dynamic programming approach to solve the SSFCMCTP exactly and efficiently. In particular, we show that our approach significantly outperforms other methods applied in the literature to solve this problem. The method exploits lower and upper bounds in order to reduce the state space variable; it is detailed in Section 2.6. We moreover preprocess the problem by applying linear programming techniques for reducing the problem size (in Section 2.5). These reduction techniques are based on ideas presented in Klose (2008) for the simpler SSFCTP and are extended here to the multiple-choice model, that is, the piecewise linear cost case. Note, however, that these extensions are not readily made, as the SSFCMCTP extends the SSFCTP in a similar manner as the multiple-choice knapsack problem extends the ordinary binary knapsack problem. In Section 2.2, we review the literature related to the SSFCMCTP. A mathematical problem formulation is introduced in Section 2.3, where an outline of the proposed solution method is also found. The LP relaxation of the problem is used to derive lower as well as upper bounds and can be solved efficiently as described in Section 2.4. The details of the dynamic programming algorithm are depicted in Section 2.6. Section 2.7 presents some computational experiences that we made with this method along with a comparison to previously posed methods. Finally, Section 2.8 summarizes the findings and concludes the paper.

## 2.2 Literature Review

The SSFCMCTP has close links with three areas of applied optimization: facility location, transportation and network flow problems and supplier selection.

In facility location, staircase cost functions are mainly considered to model the production costs (Holmberg, 1994; Holmberg and Ling, 1997; Harkness and ReVelle, 2002; Correia and Captivo, 2003). However staircase costs in transportation are only sparsely treated. Holmberg (1989) presents several decomposition approaches to solving the facility location problem with staircase transportation costs, but none of them are implemented or tested. A related problem is facility location with modular link costs. In this case, links of different sizes and costs can be established between facilities and customers. Correia, Gouveia, and Saldanha-da Gama (2010) treat this problem using a traditional and an alternative discretized model. The latter shows a stronger LP relaxation, but is pseudo-polynomial in size.

Croxton, Gendron, and Magnanti (2003b) present a network flow model with staircase transportation costs. The cost function models different modes of transportation corresponding to sending goods by *small packages*, *less-than-truckloads*, *full truckloads* and *air shipments*. Lapierre, Ruiz, and Soriano (2004) consider a similar setting, where a staircase cost structure also arises when considering shipments using the modes small packages, less-than-truckloads and full truckloads. Croxton, Gendron, and Magnanti (2003a) provide general results regarding the relative strength of different MIP formulations for *Non-Convex, Piecewise Linear Cost Minimization Problems*. Their main result shows that the LP relaxations of three standard "text-book" formulations are of equivalent strength. This includes the "multiple-choice" model that we use in Section 2.3. A special case of the above general optimization problem is the *Non-Convex, Piecewise Linear, Network Flow Problem*, where staircase cost functions are considered as an important special case. Kim and Pardalos (1999) introduce the dynamic slope scaling method for solving heuristically fixed-charge network flow problems. The procedure iteratively solves the LP relaxation and uses the resulting flows to linearize the fixed costs. Kim and Pardalos (2000) extend the method to the more general case of non-convex, piecewise linear cost functions.

The SSFCMCTP extends the SSFCTP by allowing for a staircase cost structure, instead of just a single transportation mode. Herer, Rosenblatt, and Hefter (1996) suggest the SSFCTP for modeling supplier selection problems. They propose an implicit enumeration algorithm that improves an older branch-and-bound method of Haberl (1991). A dynamic programming approach to the SSFCTP is first considered by Alidaee and Kochenberger (2005). More specifically they propose a variable transformation that reduces the worst-case complexity of a naive dynamic programming procedure. In Klose (2008), both approaches are improved and compared. The computational results obtained there indicate that the dynamic programming algorithm is more stable with respect to computation time and scales better with large problem instances. The SSFCMCTP may model multiple supplier procurement decision problems with price discounts arising in procurement and auctions. Motivated by this application, Kameshwaran and Narahari (2009) propose several heuristics for the problem along with two dynamic programming algorithms. These are based on taking the demand (as done in this paper) or the cost, respectively, as the parameter for the recursion. They base both algorithms on the fact that there is an optimal solution with at most one supplier and transportation mode supplying a positive quantity less than the capacity. For each supplier and less-than-capacity supply, they calculate the minimum cost required to meet the residual demand using a subset of the remaining suppliers and modes at full capacity. This subproblem is a *Multiple-Choice Knapsack Problem* (MCKP), which is $\mathcal{NP}$-hard. Hence, this method will, as the problem size grows, solve an increasing number of increasingly larger instances of the MCKP. In Kameshwaran and Narahari (2009) the MCKP is solved by straightforward dynamic programming, but the method might be significantly improved by using specialized code for the MCKP as the one by Pisinger (1995). Our dynamic programming algorithm does, however, directly apply to the SSFCMCTP and does not require to solve a MCKP; we only exploit the relationship of the SSFCMCTP to the MCKP when solving its LP relaxation.

Qi (2007) also uses a dynamic programming approach to solve a supplier selection problem with price breaks that are either of the all-units or the incremental type. In his model, demand depends on the product price as an additional decision variable, and total profit per period is to be maximized. For a fixed price, the problem reduces to a SSFCMCTP and may thus be solved by our method as well.

## 2.3 Problem Formulation

A single sink with a demand of $D > 0$ units can be supplied by $j = 1, \ldots, m$ retailers. The cost of serving the sink from a retailer is a non-decreasing, non-negative, left continuous, piecewise linear function (see Fig. 2.1). We denote by $q_j$ the number of transportation modes (linear segments of the cost function) available for supplier $j$ and by $f_{jl}$ the fixed cost associated with transportation mode $l$ and supplier $j$. The minimum and maximum amounts of product to be sent when using this mode are, respectively, $L_{j,l-1}$ and $L_{jl}$, where $L_{j,l-1} < L_{jl}$ and $L_{j0} = 0$. The variable cost is denoted $c_{jl} \geq 0$.

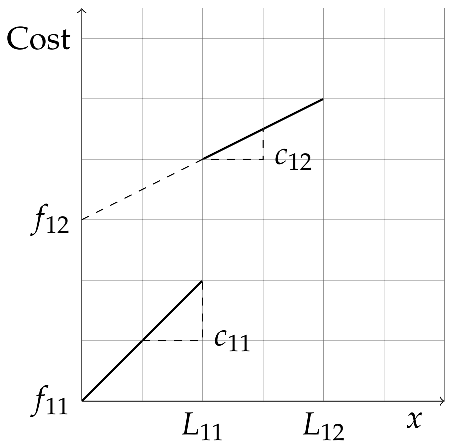



Figure 2.1: Staircase cost structure

Denoting the flow on mode $l$ from supplier $j$ by $x_{jl}$ and introducing binary variables $y_{jl}$ equal to one if mode $l$ at supplier $j$ is used, the SSFCMCTP can be formulated as the following mixed-integer linear program.

$$\text{(SSFCMCTP)} \quad z = \min \sum_{j=1}^{m} \sum_{l=1}^{q_j} (c_{jl} x_{jl} + f_{jl} y_{jl}) \tag{2.1}$$

$$\text{s.t.} \sum_{l=1}^{q_j} y_{jl} \leq 1, \quad j = 1, \ldots, m\,, \tag{2.2}$$

$$\sum_{j=1}^{m} \sum_{l=1}^{q_j} x_{jl} = D\,, \tag{2.3}$$

$$x_{jl} \leq L_{jl} y_{jl}, \quad j = 1, \ldots, m,\ l = 1, \ldots, q_j\,, \tag{2.4}$$

$$x_{jl} \geq L_{j,l-1} y_{jl}, \quad j = 1, \ldots, m,\ l = 1, \ldots, q_j\,, \tag{2.5}$$

$$x_{jl} \geq 0, \quad j = 1, \ldots, m,\ l = 1, \ldots, q_j\,, \tag{2.6}$$

$$y_{jl} \in \{0, 1\}, \quad j = 1, \ldots, m,\ l = 1, \ldots, q_j\,. \tag{2.7}$$

The objective function (2.1) is to minimize the total costs consisting of variable and fixed cost of the flow. (2.2) is the multiple-choice constraint, ensuring that at most one transportation mode is used for each supplier. The sink's demand must be met according to (2.3). Constraints (2.4) and (2.5) enforce, respectively, the upper and lower bound for each transportation mode. Finally, (2.6) and (2.7) are non-negativity and binary constraints. We assume integer-valued demand $D$ and capacities $L_{jl}$.

The solution method proposed in this paper consists of a number of steps. An outline of the entire algorithm is found below along with references to the sections detailing each step.

1. Solve the LP relaxation by the method of Section 2.4.

2. Run the primal heuristics of Subsection 2.5.1.

3. Try to reduce the problem size by the variable pegging described in Subsection 2.5.2.

4. For modes not pegged in step 3, try to strengthen the mode bounds by the method of Subsection 2.5.2.

5. Employ dynamic programming to solve the reduced problem augmented by the variable transformation of Subsection 2.6.1 and the search space reduction of Subsection 2.6.2.

## 2.4 Solving the LP Relaxation

It was shown in Croxton et al. (2003a) that solving a LP relaxation of a program like (2.1)–(2.7) is equivalent to solving the problem using the so-called "convex combination" (CC) formulation. We show here how the CC formulation reduces to the LP relaxation of a *Multiple-Choice Knapsack Problem* (MCKP). This result is easier obtained using the CC formulation, compared to using the original "multiple-choice" formulation. The LP relaxation of the CC formulation can be written as (see e.g. Croxton et al. (2003a))

$$\text{(CC-LP)} \quad z_{\text{LP}} = \min \sum_{j=1}^{m} \sum_{l=1}^{q_j} (\mu_{jl}(L_{j,l-1}c_{jl} + f_{jl}) + x'_{jl}(L_{jl}c_{jl} + f_{jl})) \tag{2.8}$$

$$\text{s.t.} \sum_{l=1}^{q_j} (\mu_{jl} + x'_{jl}) \leq 1, \quad j = 1, \ldots, m, \tag{2.9}$$

$$\sum_{j=1}^{m} \sum_{l=1}^{q_j} (\mu_{jl} L_{j,l-1} + x'_{jl} L_{jl}) = D, \tag{2.10}$$

$$\mu_{jl}, x'_{jl} \geq 0, \quad j = 1, \ldots, m, \; l = 1, \ldots, q_j. \tag{2.11}$$

The following result is readily available.

**Theorem 1**

*There exists an optimal solution in which $\mu_{jl} = 0$ for all $j$ and $l$.*

**Proof.** For the special case where $l = 1$ the non-negativity assumption on the cost function implies that $f_{j1} \geq 0$, since we defined $L_{j0} = 0$. Hence, setting $\mu_{j1} > 0$ will add a non-negative term to the objective function. It is clear that restricting $\mu_{j1} = 0$ will never change the feasibility of a solution. For the cases where $l > 1$ assume there is an optimal solution $(\mu^*_{jl}, x'^*_{jl})$ in which $\mu^*_{jl} > 0$ for some $j$ and $l$. Construct a new almost identical solution $(\overline{\mu}, \overline{x}')$

except that $\overline{\mu}_{jl} = 0$ and $\overline{x}'_{j,l-1} = x'^*_{j,l-1} + \mu^*_{jl}$. It is straightforward to see the new solution is feasible. The difference in the costs between the two solutions is

$$\begin{aligned} x'^*_{j,l-1}(L_{j,l-1}c_{j,l-1} + f_{j,l-1}) + \mu^*_{jl}(L_{j,l-1}c_{jl} + f_{jl}) - \overline{x}'_{j,l-1}(L_{j,l-1}c_{j,l-1} + f_{j,l-1}) \\ = \mu^*_{jl}(L_{j,l-1}c_{jl} + f_{jl}) - \mu^*_{jl}(L_{j,l-1}c_{j,l-1} + f_{j,l-1}) \leq 0 \end{aligned}$$

since the cost function is non-decreasing. Hence, the newly constructed solution has cost not higher than the optimal solution and therefore an optimal solution with $\mu_{jl} = 0$ for all $j$ and $l$ always exists. ■

Defining $e'_{jl} = L_{jl}c_{jl} + f_{jl}$ and adding a dummy mode for which $e'_{j0} = L_{j0} = 0$, the LP relaxation can be rewritten, using Theorem 1, as

$$\text{(LP-P)} \quad z_{\text{LP}} = \min \sum_{j=1}^{m} \sum_{l=0}^{q_j} e'_{jl} x'_{jl} \tag{2.12}$$

$$\text{s.t.} \sum_{l=0}^{q_j} x'_{jl} = 1, \quad j = 1, \ldots, m\,, \tag{2.13}$$

$$\sum_{j=1}^{m} \sum_{l=0}^{q_j} x'_{jl} L_{jl} = D\,, \tag{2.14}$$

$$x'_{jl} \geq 0, \quad j = 1, \ldots, m,\ l = 0, \ldots, q_j\,. \tag{2.15}$$

This is the LP relaxation of a MCKP. For this problem we employ the following well-known dominance criteria (see e.g. Sinha and Zoltners (1979) and Kellerer et al. (2003)):

**Definition 1**

*A variable $x'_{jl}$ is one-item dominated by $x'_{jk}$ if*

$$e'_{jl} \geq e'_{jk} \text{ and } L_{jl} \leq L_{jk}\,. \tag{2.16}$$

*Variables $x'_{ji}$ and $x'_{jk}$ two-item dominate $x'_{jl}$ if $L_{ji} < L_{jl} < L_{jk}$ and*

$$(e'_{jk} - e'_{jl})/(L_{jk} - L_{jl}) \leq (e'_{jl} - e'_{ji})/(L_{jl} - L_{ji})\,. \tag{2.17}$$

*A variable is dominated if it is dominated by one or two items.*

This can be used to reduce the number of variables in the program (LP-P), using the following theorem

**Theorem 2 (**Sinha and Zoltners (1979)**)**
*An optimal solution to the LP relaxation of the MCKP exists such that* $x'_{jl} = 0$ *for each dominated variable* $x'_{jl}$.

For each supplier $j$ the dominated modes are removed and the remaining modes denoted by $N_j$. As described in e.g. Sinha and Zoltners (1979) and Kellerer et al. (2003), this reduced problem can be solved by the following greedy algorithm, which in turn transforms the problem to an instance of the *continuous knapsack problem*:

1. Construct an instance of the binary knapsack problem's LP relaxation with objective coefficients $\tilde{e}'_{jl} = e'_{jl} - e'_{j,l-1}$ and $\tilde{L}_{jl} = L_{jl} - L_{j,l-1}$ as the weight of item $(j,l)$, where $e'_{j0} = 0$. Let the residual demand be $\tilde{D} = D$.

2. Sort the items $(j,l)$ according to non-decreasing "incremental disefficiencies" $\rho_{jl} = \tilde{e}'_{jl}/\tilde{L}'_{jl}$.

3. Use the greedy algorithm for the KP with capacity $\tilde{D}$. Initialize with $z = 0$. Each time we insert an item $(j,l)$ in the knapsack, update $z = z + \tilde{e}'_{jl}$, $\tilde{D} = \tilde{D} - \tilde{L}_{jl}$, $x'_{jl} = 1$ and $x'_{j,l-1} = 0$.

4. Let $(s,t)$ be the "split item", that is, the first item for which $\tilde{L}_{st} > \tilde{D}$ occurs when executing the above step. Set $x'_{st} = \tilde{D}/\tilde{L}_{st}$, $x'_{s,t-1} = 1 - x'_{st}$ and $z = z + \tilde{e}'_{st}x'_{st}$. Return the solution.

Alternatively one could use a more sophisticated algorithm, such as the one by Zemel (1980), which has complexity $O(mq)$ compared to the greedy algorithm's worst-case performance of $O(mq^2)$ (including the removal of dominated modes). The approach used here is similar to the one of Kameshwaran and Narahari (2009) for solving the LP relaxation. Their approach is, however, based on the inclusion of non-dominated modes instead of the exclusion of dominated modes. The complexity for both methods is $O(mq^2)$.

Before we pass the problem data to the dynamic programming algorithm, the suppliers are relabeled such that suppliers $i$ where $x_{il} = 1$ holds for exactly one $l \in N_i$ are listed first, then the split supplier $s$ and thereafter suppliers $i$ with $\sum_{l \in N_i} x'_{il} = 0$. Moreover, suppliers preceding the split supplier $s$ are relabeled in non-increasing order according to their flow in the LP solution. This is done since suppliers with a strictly positive flow in the LP solution could be seen as candidates for a good solution in the original problem.

## 2.5 Problem Reduction

Here we propose some problem reduction techniques that use Lagrangean relaxation to peg variables. Lagrangean relaxation is also used to strengthen the bounds on the flow variables (see Subsection 2.5.2).

### 2.5.1 Primal Heuristics

The knowledge of a good initial feasible solution is crucial for the performance of the problem reduction techniques. We use the following three fast and simple heuristics for obtaining such a solution. Each of the heuristics, described below, is run on the solution obtained from the LP relaxation.

- **"Actual cost" heuristic**: The flow variables from the LP solution always constitute a feasible solution to the original problem. However, in the LP relaxation the associated costs are based on the lower convex envelope of the staircase cost function. This heuristic simply uses the LP solution and recalculates the correct costs.

- **"Least cost addition" heuristic**: The fractional LP solution has the property that there is at most one supplier $s$ such that $0 < x'_{st} := x_{st}/L_{st} < 1$ and $0 < x'_{s,t-1} < 1$ for some $t \in N_s$. This heuristic takes the LP solution, but first excludes the "split supplier" $s$ by setting the corresponding flow variables to zero. We then search for a single supplier who is able to supply the amount missing to cover the total demand $D$ at lowest cost.

- **"Largest cost reduction" heuristic**: Contrary to the above procedure, set first $x_{st} = L_{st}$ (and $x_{s,t-1} = 0$ if $t - 1 \neq 0$). The supply of the resulting solution then exceeds the total demand. As long as there is excess supply, we identify the supplier yielding the largest cost reduction if his supply is reduced by the current excess (or to zero if the excess exceeds his supply).

Finally, the solution $(x^H_{jl})$ provided by one of the heuristics above could possibly be improved by solving the continuous knapsack problem

$$\min \sum_{(j,l)\in H} c_{jl} x_{jl} \tag{2.18}$$

$$\text{s.t.:} \sum_{(j,l)\in H} x_{jl} = D\,, \tag{2.19}$$

$$L_{j,l-1} \leq x_{jl} \leq L_{jl} \quad \forall\,(j,l) \in H\,, \tag{2.20}$$

where $H = \{(j,l) : x^H_{jl} > 0\}$.

### 2.5.2 Variable Pegging and Bound Strengthening

Lagrangean probing can be used to check if certain binary variables $y_{jl}$ have to be fixed to either zero or one in any solution that may improve the upper bound computed from the heuristics above (from Subsection 2.5.1). Moreover, Lagrangean relaxation and the upper bound can be exploited to possibly improve the bounds on the flow variables.

**Variable Pegging.**

Dualizing the demand constraint (2.3), in the SSFCMCTP, with multiplier $\lambda$ in a Lagrangean manner yields the Lagrangean subproblem

$$L(\lambda) = \lambda D + \min \sum_{j=1}^{m} \sum_{l=1}^{q_j} \big((c_{jl} - \lambda)x_{jl} + f_{jl}y_{jl}\big) \tag{2.21}$$

$$\text{s.t.} \sum_{l=1}^{q_j} y_{jl} \leq 1, \quad j = 1, \ldots, m\,, \tag{2.22}$$

$$x_{jl} \leq L_{jl}y_{jl}, \quad j = 1, \ldots, m,\, l = 1, \ldots, q_j\,, \tag{2.23}$$

$$x_{jl} \geq L_{j,l-1}y_{jl}, \quad j = 1, \ldots, m,\, l = 1, \ldots, q_j\,, \tag{2.24}$$

$$x_{jl} \geq 0, \quad j = 1, \ldots, m,\, l = 1, \ldots, q_j \tag{2.25}$$

$$y_{jl} \in \{0, 1\}, \quad j = 1, \ldots, m,\, l = 1, \ldots, q_j\,, \tag{2.26}$$

which can be further reduced to

$$L(\lambda) = \lambda D + \min \sum_{j=1}^{m} \sum_{l=1}^{q_j} \overline{f}_{jl}y_{jl} \tag{2.27}$$

$$\text{s.t.} \sum_{l=1}^{q_j} y_{jl} \leq 1, \quad j = 1, \ldots, m\,, \tag{2.28}$$

$$y_{jl} \in \{0, 1\}, \quad j = 1, \ldots, m,\, l = 1, \ldots, q_j\,, \tag{2.29}$$

where

$$\overline{f}_{jl} = f_{jl} + \min\{(c_{jl} - \lambda)L_{j,l-1}, (c_{jl} - \lambda)L_{jl}\}\,. \tag{2.30}$$

The above Lagrangean relaxation possesses the *integrality property*. An optimal Lagrangean multiplier $\lambda$ is thus given by constraint's (2.3) optimal dual multiplier $\lambda = \rho_{st} = \tilde{e}'_{st}/\tilde{L}_{st} = (e'_{st} - e'_{s,t-1})/(L_{st} - L_{s,t-1})$ in the LP relaxation of the SSFCMCTP. With this choice of $\lambda$, we have $L(\lambda) = z_{LP}$. Moreover, if we let

$$\overline{f}_{jl^*} = \min_{l=1,\ldots,q_j} \{\overline{f}_{jl}, 0\}\,, \tag{2.31}$$

then an optimal solution to the Lagrangean subproblem is given by

$$y_{jl} = \begin{cases} 1, \text{ if } l = l^* \text{ and } \overline{f}_{jl^*} < 0\,, \\ 0, \text{ otherwise} \end{cases} \quad \text{and} \quad x_{jl} = \begin{cases} L_{jl} y_{jl}, \text{ if } c_{jl} \le \lambda\,, \\ L_{j,l-1} y_{jl}, \text{ if } c_{jl} > \lambda\,. \end{cases} \tag{2.32}$$

Now let $z_{\mathrm{UB}}$ be the objective value of a known feasible solution. Imposing the additional constraint $y_{jl} = 1$ increases the above Lagrangean bound to

$$UP_{jl} = z_{\mathrm{LP}} - \overline{f}_{jl^*} + \overline{f}_{jl}\,.$$

Hence, if $UP_{jl} \ge z_{\mathrm{UB}}$, then we must have $y_{jl} = 0$ in any solution that improves on $z_{\mathrm{UB}}$. Similarly, enforcing $y_{jl^*} = 0$ raises the Lagrangean bound to

$$DP_{jl^*} = z_{\mathrm{LP}} - \overline{f}_{jl^*} + \min\{0, \overline{f}_{jl} : l = 1, \ldots, q_j, l \neq l^*\}\,.$$

Accordingly, if $DP_{jl^*} \ge z_{\mathrm{UB}}$, then $y_{jl^*} = 1$ holds in any solution that improves on $z_{\mathrm{UB}}$.

As with the SSFCTP, also the SSFCMCTP has an optimal solution where all, but possibly one supplier, exactly supply $L_{jl}$ (for some $l$). Thus if $H_1$ is the set of supplier and transportation mode pairs $(j, l)$ for which $y_{jl}$ could be pegged to one, we may set $x_{jl} = L_{jl}$ for each $(j, l) \in H_1$ such that $c_{jl} < \max_{(\iota,\ell)\in H_1} c_{\iota\ell}$.

**Bound Strengthening.**

The upper bound and the above Lagrangean relaxation can be further exploited to possibly strengthen the bounds on the flow variables $x_{jl}$. Let $((x^*_{jl}), (y^*_{jl}))$ denote the optimal solution (cf. (2.32)) to the Lagrangean subproblem. We can then distinguish the following cases.

**Case 1:** $x^*_{jl} = L_{jl}$. This implies that $y^*_{jl} = 1$ and thus $l = l^*$ as well as $\lambda \ge c_{jl^*}$. Setting $x_{jl^*} = B_{jl^*} < L_{jl^*}$ will then change the Lagrangean bound to

$$z_{\mathrm{LP}} - (c_{jl^*} - \lambda)(L_{jl^*} - B_{jl^*})\,.$$

Assuming that $\lambda > c_{jl^*}$ and only considering solutions improving $z_{\text{UB}}$, we obtain

$$B_{jl^*} \geq \left\lceil L_{jl^*} - (z_{\text{UB}} - z_{\text{LP}})/(\lambda - c_{jl^*}) \right\rceil =: \beta_{jl^*}\,,$$

where $\lceil a \rceil$ is the smallest integer not less than $a$. Hence, in case of $\beta_{jl^*} > 0$, we must have $y_{jl^*} = 1$ and $x_{jl^*} \geq \max\{L_{j,l^*-1}, \beta_{jl^*}\}$ in any feasible solution of objective value less than $z_{\text{UB}}$.

**Case 2:** $x_{jl} = x^*_{jl} = 0$ and $\lambda > c_{jl}$. In this case, we have $y^*_{jl} = 0$, and setting $x_{jl} = B_{jl} \in [L_{j,l-1}, L_{jl}]$ raises the Lagrangean bound to

$$z_{\text{LP}} - \overline{f}_{jl^*} + (c_{jl} - \lambda)B_{jl} + f_{jl} = UP_{jl} + f_{jl} - \overline{f}_{jl} + (c_{jl} - \lambda)B_{jl} \tag{2.33}$$

$$= UP_{jl} + (\lambda - c_{jl})(L_{jl} - B_{jl})\,, \tag{2.34}$$

where the last inequality follows from the definition of $\overline{f}_{jl}$ and $\lambda > c_{jl}$. In any solution improving the upper bound $z_{\text{UB}}$, we must thus either have $y_{jl} = 0$ or a flow $x_{jl}$ that is at least as large as

$$B_{jl} \geq \left\lceil (z_{\text{UB}} - z_{\text{LP}} + \overline{f}_{jl^*} - f_{jl})/(c_{jl} - \lambda) \right\rceil = \left\lceil L_{jl} - (z_{\text{UB}} - UP_{jl})/(\lambda - c_{jl}) \right\rceil =: \beta_{jl}\,.$$

Hence, if $L_{j,l-1} < \beta_{jl}$, the lower bound on $x_{jl}$ can be improved to $x_{jl} \geq \beta_{jl} y_{jl}$.

**Case 3:** $x_{jl} = x^*_{jl} = L_{j,l-1}$. This first implies that $\lambda < c_{jl}$ and $y^*_{jl} = 1$. Hence, $l = l^*$ and $\overline{f}_{jl} \leq 0$. Setting $x_{jl} = B_{jl}$ will change the Lagrangean bound to

$$z_{\text{LP}} + (c_{jl} - \lambda)(B_{jl} - L_{j,l-1})\,.$$

In a solution improving on $z_{\text{UB}}$, $x_{jl} = B_{jl}$ thus requires

$$B_{jl} \leq \left\lfloor L_{j,l-1} + (z_{\text{UB}} - z_{\text{LP}})/(c_{jl} - \lambda) \right\rfloor =: \beta_{jl}\,,$$

where $\lfloor a \rfloor$ is the largest integer no greater than $a$. In case of $\beta_{jl} < L_{jl}$, the upper bound on $x_{jl}$ can be strengthened.

**Case 4:** $x_{jl} = x^*_{jl} = 0$ and $\lambda < c_{jl}$. Setting $x_{jl} = B_{jl}$ in this case yields the Lagrangean bound

$$z_{\text{LP}} - \overline{f}_{jl^*} + f_{jl} + (c_{jl} - \lambda)B_{jl} = UP_{jl} - \overline{f}_{jl} + f_{jl} + (c_{jl} - \lambda)B_{jl} \tag{2.35}$$

$$= UP_{jl} + (c_{jl} - \lambda)(B_{jl} - L_{j,l-1}), \tag{2.36}$$

where the last inequality follows from the definition of $\overline{f}_{jl}$ and $\lambda < c_{jl}$. In any solution of objective value better than $z_{\text{UB}}$, we thus have either $y_{jl} = 0$ or $x_{jl} = B_{jl}$, where

$$B_{jl} \leq \left\lfloor (z_{\text{UB}} - z_{\text{LP}} + \overline{f}_{jl^*} - f_{jl})/(c_{jl} - \lambda) \right\rfloor = \left\lfloor L_{j,l-1} + (z_{\text{UB}} - UP_{jl})/(c_{jl} - \lambda) \right\rfloor =: \beta_{jl}.$$

If $\beta_{jl} < L_{jl}$, the upper bound on $x_{jl}$ can be strengthened to $x_{jl} \leq \beta_{jl} y_{jl}$.

For reasons of simplicity, we continue to denote the bounds on $x_{jl}$ by $L_{j,l-1}$ and $L_{jl}$, respectively, even though the bounds may have been improved by the means shown above.

## 2.6 Dynamic Programming Algorithm

For each supplier $j$, let $M_j$ denote the set of free variables $x_{jl}$ and $y_{jl}$ after problem reduction. Define

$$\begin{aligned} G_i(S) = \min \sum_{j=1}^{i} \sum_{l \in M_j} (c_{jl} x_{jl} + f_{jl} y_{jl}) \quad & \\ \text{s.t.} \sum_{j=1}^{i} \sum_{l \in M_j} x_{jl} = S\,, & \\ \sum_{l \in M_j} y_{jl} \leq 1, \quad j = 1, \dots, i\,, & \\ L_{j,l-1} y_{jl} \leq x_{jl} \leq L_{jl} y_{jl}, \quad j = 1, \dots, i,\ l \in M_j\,, & \\ y_{jl} \in \{0, 1\}, \quad j = 1, \dots, i,\ l \in M_j\,, & \\ x_{jl} \geq 0, \quad j = 1, \dots, i,\ l \in M_j & \end{aligned} \tag{2.37}$$

for $S = S_i^{\min}, \dots, S_i^{\max}$, where

$$S_i^{\min} = \max\Big\{0, D - \sum_{j=i+1}^{m} L_{jq_j}\Big\} \quad \text{and} \quad S_i^{\max} = \min\Big\{D, \sum_{j=1}^{i} L_{jq_j}\Big\}.$$

The function $G_i(S)$ can be computed recursively as

$$G_i(S) = \min\{G_{i-1}(S), \min_{l \in M_i} T_{il}(S)\}, \tag{2.38}$$

where

$$T_{il}(S) = \min\{f_{il} + c_{il}x_{il} + G_{i-1}(S - x_{il}) : x_{il} = L_{i,l-1}, \dots, x_{il}^{\max} := \min\{L_{il}, S\}\}. \tag{2.39}$$

When $S \notin [S_i^{\min}, S_i^{\max}]$, we let $G_i(S) = \infty$. Moreover, $G_m(D) = z^*$.

### 2.6.1 Variable Transformation

Using the recursion (2.38) to compute $G_m(D)$ requires in the worst case a computational effort of $O(Dmqk_{\max})$, where $q = \max_j q_j$ and $k_{\max} = \max_{(j,l)}\{L_{jl} - L_{j,l-1} + 1\}$. For the SSFCTP, Alidaee and Kochenberger (2005) propose a variable transformation that allows to reduce this worst-case effort to $O(Dmq)$.

Inserting (2.39) into (2.38) gives

$$G_i(S) = \min\{G_{i-1}(S), \min_{l \in M_i} \min\{f_{il} + c_{il}x + G_{i-1}(S - x) : \ x = L_{i,l-1}, \dots, L_{il}\}\}.$$

Using

$$K_{il}(r) = G_{i-1}(r) - c_{il}r,$$

the recursion can be rewritten as

$$G_i(S) = \min\{G_{i-1}(S), \min_{l \in M_i}\{f_{il} + c_{il}S + \min_r K_{il}(r) : \ r = S - L_{il}, \dots, S - L_{i,l-1}\}\}. \tag{2.40}$$

We implement this by keeping a sorted list

$$KList_l = \{(r_1, K_{il}(r_1)), (r_2, K_{il}(r_2)), \dots, (r_K, K_{il}(r_K))\}$$

of pairs $(r, K_{il}(r))$ for each transportation mode $l$, where $r_k$ is the largest $r$ such that $K_{il}(r) < K_{il}(r_{k+1})$. Also here dominance is relevant. A state $(r, K_{il}(r))$ dominates a state $(r', K_{il}(r'))$ if $r > r'$ and $K_{il}(r) < K_{il}(r')$. At the point where the dynamic programming algorithm reaches supplier $i$, the $KList_l$'s are computed for $S = S_i^{\min}$ and then updated accordingly when going from $S$ to $S+1$. The recursion can then be simplified to

$$G_i(S) = \min\{G_{i-1}(S), \min_{l \in M_i}\{f_{il} + c_{il}S + K_{il}(r_1)\}\} .$$

Note that $r_1$ depends on $i$ and $l$. As Alidaee and Kochenberger (2005) show, for a given $i$ and $l$, the total effort for updating the list is bounded by $O(D)$. Hence, the complete dynamic programming procedure can be carried out in $O(Dqm)$ time.

### 2.6.2 Search Space Reduction

One of the heavy computational tasks of this algorithm is to determine $G_i(S)$ for all feasible values of $S$ (those in the interval $[S_i^{\min}, S_i^{\max}]$). A significant speed-up can be expected if this interval is reduced in some way. For the SSFCTP, this is done by Klose (2008), obtaining stronger bounds on $S_i^{\min}$. Here bounds for the SSFCMCTP are presented using a similar methodology.

For a given $S$, consider the program

$$\begin{aligned} \overline{G}_i(S) = \min & \sum_{j=i+1}^{m} \sum_{l \in M_j} (c_{jl}x_{jl} + f_{jl}y_{jl}) \\ \text{s.t.} & \sum_{j=i+1}^{m} \sum_{l \in M_j} x_{jl} = D - S\,, \\ & \sum_{l \in M_j} y_{jl} \leq 1, \quad j = i+1, \ldots, m\,, \\ & L_{j,l-1}y_{jl} \leq x_{jl} \leq L_{jl}y_{jl}, \quad j = i+1, \ldots, m,\, l \in M_j\,, \\ & x_{jl} \geq 0,\, y_{jl} \in \{0,1\}, \quad j = i+1, \ldots, m,\, l \in M_j\,. \end{aligned} \tag{2.41}$$

Hence, $G_i(S) + \overline{G}_i(S)$ describes the total minimum cost that needs to be spent if the first $i$ suppliers supply an amount of $S$. If $G_i(S) + \overline{G}_i(S) \geq z_{\text{UB}}$, then state $S$ can be discarded. In particular, if $z_{i1}(S)$ and $z_{i2}(S)$ denote lower bounds on $G_i(S)$ and $\overline{G}_i(S)$ respectively, then this is already true if $z(S) := z_{i1}(S) + z_{i2}(S) \geq z_{\text{UB}}$. Lower bounds $z_{i1}(S)$ and $z_{i2}(S)$ can be derived from the LP relaxation of (2.37) and (2.41), that is, from the linear programs

$$\begin{aligned} z_{i1}(S) = \min & \sum_{j=1}^{i} \sum_{l \in N_j} e'_{jl} x'_{jl} \\ \text{s.t.} & \sum_{l \in N_j} x'_{jl} \leq 1, \quad j = 1, \ldots, i, \\ & \sum_{j=1}^{i} \sum_{l \in N_j} x'_{jl} L_{jl} = S, \\ & x'_{jl} \geq 0, \quad j = 1, \ldots, i, \ l \in N_j, \end{aligned} \tag{2.42}$$

and

$$\begin{aligned} z_{i2}(S) = \min & \sum_{j=i+1}^{m} \sum_{l \in N_j} e'_{jl} x'_{jl} \\ \text{s.t.} & \sum_{l \in N_j} x'_{jl} \leq 1, \quad j = i+1, \ldots, m, \\ & \sum_{j=i+1}^{m} \sum_{l \in N_j} x'_{jl} L_{jl} = D - S, \\ & x'_{jl} \geq 0, \quad j = i+1, \ldots, m, \ l \in N_j. \end{aligned} \tag{2.43}$$

Recall that $N_j$ is the set of non-dominated items. Evaluating the linear programs as a function of $S$ is, however, still too much of an effort. Thus, we work again with lower bounds on $z_{i1}(S)$ and $z_{i2}(S)$ by taking the dual of the linear programs above and deriving dual feasible solutions. The duals have the following structure

$$z_{i1}(S) = \max S\sigma_1 - \sum_{j=1}^{i} \eta_j \tag{2.44}$$

$$\text{s.t. } \sigma_1 L_{jl} - \eta_j \leq e'_{jl}, \quad j = 1, \ldots, i, \ l \in N_j, \tag{2.45}$$

$$\sigma_1 \in \mathcal{R}, \ \eta_j \geq 0, \quad j = 1, \ldots, i, \tag{2.46}$$

and

$$z_{i2}(S) = \max\ (D - S)\sigma_2 - \sum_{j=i+1}^{m} \eta_j \tag{2.47}$$

$$\text{s.t. } \sigma_2 L_{jl} - \eta_j \leq e'_{jl}, \quad j = i+1, \ldots, m,\ l \in N_j\,, \tag{2.48}$$

$$\sigma_2 \in \mathcal{R},\ \eta_j \geq 0, \quad j = i+1, \ldots, m\,. \tag{2.49}$$

Both programs are strongly connected to the LP relaxation's (LP-P) dual program that can be stated as

$$z_{\mathrm{LP}} = \max\ D\sigma - \sum_{j=1}^{m} \eta_j \tag{2.50}$$

$$\text{s.t. } \sigma L_{jl} - \eta_j \leq e'_{jl}, \quad j = 1, \ldots, m,\ l \in N_j\,, \tag{2.51}$$

$$\sigma \in \mathcal{R},\ \eta_j \geq 0, \quad j = 1, \ldots, m\,. \tag{2.52}$$

Having solved the primal problem, (LP-P), one can easily find the optimal dual solution using complementary slackness conditions. Let $(\sigma^*, (\eta_j^*))$ denote this optimal dual solution.

Again let $s$ denote the split supplier, that is, the one supplying a positive amount less than the upper limit at the employed transportation mode $t$. Recall that the suppliers $i$ are arranged in such a way that in the LP solution we have $\sum_{l \in N_i} x'_{il} = x'_{il_i} = 1$ for some $l_i \in N_i$ in case of $i < s$ and $\sum_{l \in N_i} x'_{il} = 0$ if $i > s$. We distinguish the following two cases.

**Case 1, $i < s$.** Let $\overline{\sigma}_i = \rho_{il_i}$ be the incremental disefficiency of supplier $i$ at transportation mode $l_i$. A feasible solution to (2.44) is to set

$$\sigma_1 = \overline{\sigma}_i \quad \text{and} \quad \eta_j = \overline{\eta}_j^i := \max_{l \in N_j} \{L_{jl}\overline{\sigma}_i - e'_{jl}, 0\} \text{ for } j = 1, \ldots, i\,.$$

A feasible solution to (2.47) is readily available by letting $\sigma_2 = \sigma^*$ and $\eta_j = \eta_j^*$ for $j = i+1,\ldots,m$. The lower bound on $z_{i1}(S) + z_{i2}(S)$ obtained in this way is then

$$\overline{\sigma}_i S - \sum_{j=1}^{i} \overline{\eta}_j^i + \sigma^*(D-S) - \sum_{j=i+1}^{m} \eta_j^* = \sigma^* D - \sum_{j=1}^{m} \eta_j^* + (\overline{\sigma}_i - \sigma^*)S + \sum_{j=1}^{i} (\eta_j^* - \overline{\eta}_j^i) \tag{2.53}$$

$$= z_{\mathrm{LP}} + \sum_{j=1}^{i} (\eta_j^* - \overline{\eta}_j^i) - (\sigma^* - \overline{\sigma}_i)S\,. \tag{2.54}$$

Note that $\overline{\sigma}_i = \rho_{il_i} \leq \sigma^* = \rho_{st}$ and thereby also $\overline{\eta}_j^i \leq \eta_j^* \;\forall\, j \leq i$, since $i < s$. State $S$ may then only lead to an improved solution if

$$z_{\mathrm{LP}} + \sum_{j=1}^{i} (\eta_j^* - \overline{\eta}_j^i) - (\sigma^* - \overline{\sigma}_i)S \leq z_{\mathrm{UB}}\,. \tag{2.55}$$

In case of $\overline{\sigma}_i < \sigma^*$, we thus obtain

$$S \geq \left\lceil \Big( \sum_{j=1}^{i} (\eta_j^* - \overline{\eta}_j^i) - z_{\mathrm{UB}} + z_{\mathrm{LP}} \Big) / (\sigma^* - \overline{\sigma}_i) \right\rceil . \tag{2.56}$$

**Case 2,** $i \geq s$**.** If $i > s$, we have $x_{il} = 0 \;\forall\, l \in N_i$ in the LP solution and thus $\eta_i^* = 0$. A feasible solution to (2.44) is to simply set

$$\sigma = \sigma^* \text{ and } \eta_j = \eta_j^* \text{ for } j = 1,\ldots,i\,,$$

which yields

$$\sigma^* S - \sum_{j=1}^{i} \eta_j^* = \sigma^* D - \sigma^*(D-S) - \sum_{j=1}^{m} \eta_j^* = z_{\mathrm{LP}}^* - \sigma^*(D-S) \tag{2.57}$$

as a lower bound on $z_{i1}(S)$.

A feasible solution to (2.47) is found by setting $\eta_j = \eta_j^* = 0$ for $j = i+1,\ldots,m$ and

$$\sigma_2 = \tilde{\sigma}_i := \min_{j=i+1,\ldots,m,\, l \in N_j} \left\{ e'_{jl} / L_{jl} \right\}.$$

This gives

$$(D-S)\tilde{\sigma}_i + \sum_{j=i+1}^{m} \eta_j^* = (D-S)\tilde{\sigma}_i \tag{2.58}$$

as a lower bound on $z_{i2}(S)$, and in total

$$z^*_{\mathrm{LP}} - \sigma^*(D-S) + (D-S)\tilde{\sigma}_i = z^*_{\mathrm{LP}} + (\tilde{\sigma}_i - \sigma^*)(D-S)$$

as a lower bound on $z_i(S) = z_{i1}(S) + z_{i2}(S)$. Accordingly, state $S$ may only lead to an improved solution if

$$z^*_{\mathrm{LP}} + (\tilde{\sigma}_i - \sigma^*)(D-S) \le z_{\mathrm{UB}}\,. \tag{2.59}$$

Note that in the optimal solution $(\sigma^*, (\eta^*_j))$ to (2.50) we have

$$\sigma^* \le \min_{j=s+1,\ldots,m,\, l_j \in N_j} \left\{ (e'_{jl} + \eta^*_j)/L_{jl} \right\} = \min_{j=s+1,\ldots,m,\, l_j \in N_j} \left\{ e'_{jl}/L_{jl} \right\} \le \tilde{\sigma}_i\,.$$

Inequality (2.59) thus implies that

$$S \ge \left\lceil D - (z_{\mathrm{UB}} - z_{\mathrm{LP}})/(\tilde{\sigma}_i - \sigma^*) \right\rceil$$

provided that $\tilde{\sigma}_i > \sigma^*$. Again, this might be used to obtain a stronger bound on $S_i^{\min}$.

## 2.7 Computational Experience

We tested the solution method's performance on a number of randomly generated test problem instances. As a benchmark, we solved the same instances using CPLEX's MILP solver (version 12.1).

### 2.7.1 Generation of Test Instances

Holmberg (1994) investigates facility location problems with staircase production costs and proposes a method for generating test data. This method is employed here with the extension that the variable cost can be non-zero. Starting with $L_{j0} = 0$, the increase $L_{jl} - L_{j,l-1}$ in capacity is $U[k, 2k]$ (and integer), where $k = \lfloor 100n/(mq) \rfloor$ and $U[x, y]$ is a uniformly distributed random number on the range $[x, y]$. In Holmberg's application, $n$ is the number of customers, which in this case is one. Since this is not meaningful, we disaggregated the single sink into three possible values $n \in \{500, 1000, 1500\}$. The demand was then calculated as $D = 60 \cdot n$ (the expected value of the total demand in Holmberg (1994)). The number of transportation modes was chosen from $q \in \{3, 5, 10\}$ and the number of suppliers $m \in \{500, 1000, 5000, 10000\}$. The number of modes was fixed for all suppliers in the instances generated. The increase $f_{jl} - f_{j,l-1}$ in fixed cost was drawn from $U[10k, 40k]$. The variable cost $c_{jl}$ was calculated as $c_{jl} = (f_{jl} - f_{j,l-1})e/(L_{jl})$. Three different kinds of variable costs were considered: a random cost case where $e$ was drawn from $U[0, 1]$, a full cost case where $e = 1$ and a zero cost case where $e = 0$.

The ratio between total demand and total capacity is often highlighted as an important factor of difficulty. In order to analyze the effect, we modified a subset of the test instances by increasing the default demand by

$$(\text{Total capacity} - \text{Demand})\sigma, \ \sigma \in \{0.4, 0.6, 0.8\} \tag{2.60}$$

and rounding off to the nearest integer. The total capacity was defined as the sum of the $L_{jq}$, hence the maximum amount that could be sent from all suppliers. In the actual instances, the increase in demand was substantial, in general roughly doubling the demand. Table 2.1 gives an overview of the parameters used in the generation of instances.

Table 2.1: A summary of parameters

| | |
|---|---|
| Number of retailers ($m$) | 500,1000,5000,10000 |
| Default demand ($D$) | 30000,60000,90000 |
| Increase of demand ($\sigma$) | 0.0, 0.4, 0.6, 0.8 |
| Variable cost ($e$) | $U$[0,1] (Random), 0 (Zero), 1 (Full) |
| Number of modes ($q$) | 3, 5, 10 |

Additionally, we have generated instances using the generator proposed by Kameshwaran and Narahari (2009). For these instances we set the demand to 40% of the total supply. These instances are divided into instances with small modes (DP-1) and large modes (DP-2). For each configuration, we generated 5 instances, hence, a total of 240 instances were solved. The largest instances have $10,000 \cdot 10 = 100,000$ binary variables. Each instance was solved five times. The minimum and maximum values obtained for the CPU time were then removed and the remaining three averaged. The CPU times displayed in the subsequent tables are averages of the so measured CPU time of a single instance. A limit of 30 mins (1800 secs) were imposed for all algorithms, in which case the algorithm was halted and the time limit used as the averages.

### 2.7.2 Computational Results

The experiments were conducted on a laptop with 3 GB RAM and an Intel Core 2 Duo 2.4 GHz processor. All algorithms were executed on a single processor. CPLEX was run with default parameters, except for the limitation to one processor and a required relative optimal tolerance of $10^{-7}$. With this tolerance, the instances showing the largest total cost were solved with an accuracy to the first decimal. The reported CPU times exclude the time required for input and output operations. For comparison, we also tested the dynamic programming method of Kameshwaran and Narahari (2009). It is based on the fact that in the optimal solution there will be at most one retailer, supplying less than mode capacity, i.e. at most one where $L_{j,l-1} < x_{jl} < L_{jl}$. This can be exploited by calculating the minimum cost for each retailer and less-than full capacity supply, where the residual demand is supplied by a subset of the remaining retailers. These retailers are in turn restricted to only full capacity flows. The subproblem of calculating the minimum cost for the remaining retailers is a MCKP. The method proposed in Kameshwaran and Narahari (2009) uses a simple dynamic programming approach to solve this, but we augment this by the method of Pisinger (1995). Preliminary results indicated this improved the performance of the algorithm radically. This method will be denoted KNP. The dynamic programming algorithm proposed in this paper will be denoted by CKA. Note that the CKA method has a worst-case performance of $O(mqD)$ compared to a worst case of $O(m^2qD)$ for the KNP method.

Table 2.2 shows the performance of all algorithms on the instances generated as in Kamesh-

Table 2.2: CPU times (secs) for instances generated as in Kameshwaran and Narahari (2009)

| $m$ | DP | CKA (min, max) | CPLEX (min, max) | KNP (min, max) |
|---|---|---|---|---|
| 500 | 1 | 0.01 (0.00, 0.01) | 0.82 (0.34, 1.33) | 1.47 (1.34, 1.73) |
| | 2 | 0.04 (0.02, 0.07) | 0.34(0.18, 0.57) | 15.30 (4.73, 22.85) |
| 1000 | 1 | 0.02 (0.00, 0.04) | 0.85 (0.38, 1.07) | 2.36 (1.90, 2.80) |
| | 2 | 0.08 (0.01, 0.21) | 0.64(0.39, 1.34) | 8.52 (3.77, 13.52) |
| 5000 | 1 | 0.15 (0.10, 0.20) | 9.11 (5.53, 19.82) | 115.73 (107.52, 137.31) |
| | 2 | 0.49 (0.03, 0.87) | 274.19 (2.80, 680.01) | 178.04 (85.65, 301.87) |
| 10000 | 1 | 0.96 (0.20, 1.37) | 1465.82 (1346.35, 1602.38) | 762.81(474.86, 960.37) |
| | 2 | 0.88 (0.16, 2.27) | 736.51 (22.04, 1800) | 831.84 (401.62, 1215.97) |

waran and Narahari (2009). Comparing the KNP method to CPLEX it seems heavily dependent on the setting which method is the fastest. Both are, however, significantly slower than the CKA method.

Table 2.3 shows the performance of the algorithms on a number of standard instances generated as described in Subsection 2.7.1 (where $\sigma = 0$, $q = 5$ and the variable cost is random). The worst performance of the CKA algorithm is on the smaller test instances. A likely reason for this is a larger duality gap (i.e., the gap between the LP relaxation and the optimal (integer) solution) compared to that on the larger instances. For all configurations of $m$, there is a negative effect on the performance for larger demand and strongest on the smallest and largest instances. Profiling the CKA code showed that most of the computational effort is spent on the pure dynamic programming part (see Section 2.6), as expected. On a particular instance this accounted for 98% of the total solution time.

For the remainder of the test we have focused on the effect of the parameters on the CKA algorithm as well as CPLEX. Table 2.4 shows the influence of the number of modes when $D = 60000$, the variable cost is random and $\sigma = 0$.

As could be expected, the instances become more difficult to solve if $q$ increases. For CPLEX, the required computation time increases overproportionately with $q$. For the CKA algorithm, this does not seem to be the case, in particular if one takes into account that the averages in column 3 of Table 2.4 are slightly "biased": For each configuration there is

Table 2.3: CPU times (secs) for standard instances

| $m$ | $D$ | CKA (min, max) | CPLEX (min, max) | KNP (min, max) |
|---|---|---|---|---|
| 500 | 30000 | 0.05 (0.01, 0.10) | 1.96 (0.40, 3.00) | 1.04 (0.90, 1.16) |
| | 60000 | 0.11 (0.01, 0.48) | 1.66 (0.40, 4.15) | 1.09 (0.99, 1.24) |
| | 90000 | 0.17 (0.05, 0.40) | 2.88 (0.93, 7.96) | 1.21 (1.05, 1.43) |
| 1000 | 30000 | 0.04 (0.01, 0.09) | 4.66 (0.65, 7.68) | 4.43 (4.25, 4.56) |
| | 60000 | 0.04 (0.02, 0.08) | 3.63 (1.31, 7.87) | 4.46 (3.88, 5.02) |
| | 90000 | 0.05 (0.02, 0.09) | 3.49 (1.69, 5.08) | 4.16 (3.87, 4.84) |
| 5000 | 30000 | 0.04 (0.02, 0.05) | 31.38 (12.82, 42.30) | 110.61 (91.28, 154.51) |
| | 60000 | 0.05 (0.02, 0.06) | 30.58 (22.58, 36.77) | 122.52 (111.83, 161.90) |
| | 90000 | 0.06 (0.05, 0.07) | 40.96 (27.49, 61.05) | 150.51 (136.48, 165.07) |
| 10000 | 30000 | 0.07 (0.04, 0.11) | 118.19 (1.96, 302.16) | 418.04 (286.68, 477.70) |
| | 60000 | 0.09 (0.04, 0.11) | 108.10 (62.20, 152.85) | 466.84 (441.08, 487.68) |
| | 90000 | 0.11 (0.10, 0.11) | 193.34 (147.93, 224.89) | 780.80 (734.92, 884.62) |

Table 2.4: CPU times (secs) for influence of modes

| $m$ | $q$ | CKA (min, max) | CPLEX (min, max) |
|---|---|---|---|
| 500 | 3 | 0.02 (0.01, 0.03) | 2.8 (0.43, 5.61) |
| | 5 | 0.11 (0.01, 0.48) | 1.66 (0.40, 4.15) |
| | 10 | 0.07 (0.02, 0.11) | 7.92 (1.86, 13.75) |
| 1000 | 3 | 0.02 (0.01, 0.06) | 2.19 (1.49, 3.19) |
| | 5 | 0.04 (0.02, 0.08) | 3.63 (1.31, 7.87) |
| | 10 | 0.11 (0.03, 0.24) | 9.40 (3.24, 24.02) |
| 5000 | 3 | 0.04 (0.04, 0.05) | 14.38 (11.42, 19.63) |
| | 5 | 0.05 (0.02, 0.06) | 30.58 (22.58, 36.77) |
| | 10 | 0.08 (0.04, 0.10) | 88.14 (1.92, 127.05) |
| 10000 | 3 | 0.19 (0.03, 0.77) | 48.64 (0.85, 83.16) |
| | 5 | 0.09 (0.04, 0.11) | 108.10 (62.20, 152.85) |
| | 10 | 0.12 (0.09, 0.24) | 193.23 (4.59, 374.08) |

always just a single instance proving to be relatively hard and requiring the largest CPU time (e.g., in case of $m = 500$ and $q = 5$, the minimum CPU time is 0.01 whereas the maximum is 0.48).

The impact of the gap between supply and demand on the computational performance is summarized in Table 2.5. Again, $D = 60000$, the variable cost is random and $q = 5$. Although the CKA algorithm's theoretical worst-case performance of $O(Dmq)$ time suggests

Table 2.5: CPU times (secs) for influence of tightness

| $m$ | $\sigma$ | CKA | CPLEX |
|---|---|---|---|
| 500 | 0.0 | 0.11 (0.01, 0.48) | 1.66 (0.40, 4.15) |
| | 0.4 | 0.14 (0.02, 0.26) | 7.18 (0.98, 16.51) |
| | 0.6 | 0.15 (0.02, 0.34) | 3.39 (1.88, 4.74) |
| | 0.8 | 0.25 (0.08, 0.72) | 3.16 (1.49, 6.07) |
| 1000 | 0.0 | 0.04 (0.02, 0.08) | 3.63 (1.31, 7.87) |
| | 0.4 | 0.06 (0.03, 0.10) | 4.48 (2.08, 6.88) |
| | 0.6 | 0.04 (0.02, 0.05) | 7.35 (1.58, 15.87) |
| | 0.8 | 0.05 (0.01, 0.14) | 6.08 (4.45, 8.91) |
| 5000 | 0.0 | 0.05 (0.02, 0.06) | 30.58 (22.58, 36.77) |
| | 0.4 | 0.04 (0.02, 0.05) | 58.69 (26.64, 92.49) |
| | 0.6 | 0.20 (0.04, 0.86) | 72.53 (29.26, 127.39) |
| | 0.8 | 0.55 (0.04, 2.58) | 114.08 (71.87, 137.57) |
| 10000 | 0.0 | 0.09 (0.04, 0.11) | 108.10 (62.20, 152.85) |
| | 0.4 | 0.22 (0.04, 0.91) | 207.98 (117.55, 333.41) |
| | 0.6 | 0.06 (0.03, 0.08) | 400.82 (327.03, 591.51) |
| | 0.8 | 0.04 (0.03, 0.07) | 384.61 (155.55, 706.65) |

that the computational effort increases proportionately with $D$, this can only be observed sporadically for our implementation and the tested instances (e.g. for $m = 500$, but not for $m = 10000$). It seems that the applied problem and state space reductions are successful in dampening the demand's impact on the computation time.

For the SSFCTP, computational performance is often dependent on the relative size of the variable cost compared to the fixed cost. In order to check this, we also looked at the full and zero cost case for the test instances with $\sigma = 0$, $D = 60000$ and $q = 5$. Table 2.6 shows that the random cost case seems the easiest. The CKA algorithm is slowest on the zero cost instances. This possibly stems from the fact that in this case linearizing the fixed cost causes larger duality gaps, which in turn results in less extensive problem size and state space reductions. For CPLEX, the zero cost case also appears to be the most difficult.

In summary, the dynamic programming algorithm is proven to be rather fast and relatively stable over the range of parameters tested.

Table 2.6: CPU times (secs) for influence of variable cost

| $m$ | Cost | CKA | CPLEX |
|---|---|---|---|
| 500 | R | 0.11 (0.01, 0.48) | 1.66 (0.40, 4.15) |
| | Z | 0.15 (0.01, 0.51) | 3.40 (1.31, 5.09) |
| | F | 0.10 (0.02, 0.24) | 2.37 (1.91, 3.48) |
| 1000 | R | 0.04 (0.02, 0.08) | 3.63 (1.31, 7.87) |
| | Z | 0.12 (0.01, 0.27) | 3.95 (1.15, 7.88) |
| | F | 0.05 (0.01, 0.11) | 5.31 (2.49, 8.24) |
| 5000 | R | 0.05 (0.02, 0.06) | 30.58 (22.58, 36.77) |
| | Z | 0.06 (0.02, 0.07) | 74.79 (0.68, 139.82) |
| | F | 0.05 (0.05, 0.06) | 25.88 (15.71, 48.39) |
| 10000 | R | 0.09 (0.04, 0.11) | 108.10 (62.20, 152.85) |
| | Z | 0.26 (0.07, 0.98) | 255.34 (114.57, 413.06) |
| | F | 0.06 (0.02, 0.07) | 61.20 (1.82, 113.38) |

## 2.8 Conclusion

In this paper, the Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem has been treated. This model is appropriate in many real world applications such as, for instance, supplier selection. To solve the problem to optimality, a dynamic programming algorithm is proposed. The basic procedure is improved using variable pegging and a state space reduction method. Our computational results show that this solution method is significantly faster than other methods proposed in the literature. The tests also indicate that the algorithm can provide almost instantaneous solutions, promoting its use, for instance, within e-procurement software systems that aim at automatizing procurement and supplier selection decisions. Moreover, the code is so fast that it might be used as a subroutine in more complicated settings, e.g., as a pricing routine within a column generation scheme.

# Chapter 3

# Speed-up Techniques for the Capacitated Facility Location Problem with Piecewise Linear Transportation Costs

# Speed-up Techniques for the Capacitated Facility Location Problem with Piecewise Linear Transportation Costs

Tue R. L. Christensen

Department of Economics and Business, Aarhus University, Denmark,
tuec@asb.dk

---

**Abstract**

In this paper we consider a capacitated facility location problem with a piecewise linear transportation cost structure and consider two mixed-integer formulations of the problem. We investigate various procedures for speeding up the solution time of this problem when using a standard mixed-integer programming solver. These procedures include the addition of both known and new valid inequalities as well as a Lagrangean heuristic. The Lagrangean heuristic provides tight bounds on the problem in very short computation time and can also be used to reduce the problem size by pegging variables. Computational tests show that the valid inequalities considerably tighten the linear programming relaxation of the models for both formulations. In total one can achieve a reduction in the total computation time of up to 80% by using the speed-up techniques proposed in this paper.

---

## 3.1 Introduction

In this paper we consider a capacitated facility location problem (CFLP) with piecewise linear transportation costs (PLTC) and consider a number of valid inequalities as well as a Lagrangean heuristic for the problem to speed up the solution process. Standard facility location problems assume that the transportation costs are directly proportional to the amount shipped, but this ignores several important real-world aspects for a variety of applications. Often, transportation costs have a piecewise linear, non-decreasing structure due to the different transportation modes used, modes such as *small packages, less-than-truckloads,* and *full truckloads* (Croxton et al. (2003b)). This kind of cost structure can be found in many real-life industries such as the logistics industry (Lapierre et al. (2004)) and in the shipping industry (Baumgartner et al. (2012)). Additionally, price discounts such as all-unit or incremental discounts are often modeled on the basis of this cost structure (see Kameshwaran and Narahari (2009), Christensen et al. (2012)). Ignoring such important issues may lead to a suboptimal facility layout and a resulting loss for the company.

In Section 3.2 we review the literature related to the capacitated facility location problem with piecewise linear transportation costs. The problem is described in Section 3.3 and two mathematical formulations are given, namely the *discretized model* and the *multiple-choice model*, and we also show that both models provide the same linear programming relaxation. In Section 3.4 we review the valid inequalities proposed by Correia et al. (2010) for the *discretized model* and derive similar inequalities for the *multiple-choice model*. We propose a Lagrangean heuristic which is described in detail in Section 3.5. Computational results are presented in Section 3.6 and a conclusion is given in Section 3.7.

## 3.2 Related Works

The related literature for the capacitated facility location problem with piecewise linear transportation costs can be roughly divided into articles addressing: facility location with convex/concave/modular transportation costs, facility location with piecewise linear production costs, and transportation problems with a piecewise linear cost function.

Very few articles consider nonlinear transportation costs in a facility location context. Holmberg (1999) assumes convex transportation costs and then linearizes this function by introducing a breakpoint for each integer flow for the uncapacitated facility location problem. Correia et al. (2010) consider a capacitated facility location problem where the capacity on the arcs used must be decided by selecting a combination of modules. In Section 3.A we show that the capacitated facility location problem with modular transportation costs can be reformulated as a capacitated facility location problem with piecewise linear transportation costs. For the modular problem Correia et al. (2010) propose a discretized reformulation and a number of valid inequalities for this model which theoretically and computationally outperforms their initial formulation of the problem. The discretized model is also applicable for the problem considered in this paper and we revisit the model in Subsection 3.3.2 and the valid inequalities proposed for the model in Subsection 3.4.1. Soland (1974) considers a continuous concave transportation cost structure which is then approximated by a piecewise linear function. This approximation is then iteratively improved by a branching scheme to obtain an exact solution method. Lapierre et al. (2004) considers the problem of placing transshipment hubs using piecewise linear transportation costs and develop several heuristics for the problem. Baumgartner et al. (2012) propose a number of heuristics for a 3-echelon, multi-product, uncapacitated supply chain problem with piecewise linear transportation costs faced by a chemical company.

While few articles consider piecewise linear transportation costs, we have found a rather ample body of literature on the capacitated facility location problem with a piecewise linear production cost function. This includes works on Lagrangean heuristics by Kubo and Kasugai (1991); Holmberg and Ling (1997); Harkness and ReVelle (2002); Correia and Captivo (2003); Broek et al. (2006); Correia and Captivo (2006) and on exact methods by Holmberg (1994), and Dasci and Verter (2001). More general cost functions have also been considered in e.g. Mirchandani and Jagannathan (1989), and Wu et al. (2006) and piecewise linear convex production costs in Harkness and ReVelle (2003).

Finally, a number of articles have been published on the pure transportation part of the problem, termed the *piecewise linear network flow problem* (PLNFP). Lamar (1993) shows how to transform any nonlinear arc cost network into a network with concave costs at the expense of an increase in the number of arcs. Kim and Pardalos (2000) present a heuristic for the PLNFP based on iteratively linearizing the cost function and solving a series of simple transportation problems. We utilize this heuristic in the Lagrangean heuristic in this paper and provide a detailed description in Subsection 3.5.2. Croxton et al. (2003a) show that several standard formulations for representing a piecewise linear function provide the same linear programming bound. Vielma et al. (2010) extend this result by including a number of new models and they also consider non-separable piecewise linear cost functions. The latter paper presents computational results that strongly indicate that the multiple-choice model formulated in Subsection 3.3.1 is a superior formulation when using the standard solver CPLEX for instances where the number of modes is less than 16 in terms of total time used to compute the optimal solution. As neither Croxton et al. (2003a) nor Vielma et al. (2010) consider the discretized model proposed in Correia et al. (2010) we comment on their work by noting that it is yet another model whose LP relaxation approximates the piecewise linear function by the lower convex envelope, like e.g. the multiple-choice model.

## 3.3 Problem Description

The capacitated facility location problem with piecewise linear transportation costs can be stated as follows. Define $I = \{1, \ldots, n\}$ as the set of potential facility sites and the set of customers as $J = \{1, \ldots, m\}$. Each customer has a demand of $d_j$ units. Each facility site has an associated capacity of $S_i$ units and a fixed setup cost of $f_i$. The cost of transporting goods between each facility and customer follows a non-decreasing piecewise linear cost function and each line segment of the function is known as a mode. Let $Q_{ij} = \{1, \ldots, q_{ij}\}$ denote the set of modes between facility $i$ and customer $j$. For ease of notation, we assume that $q_{ij} = q$, for all $(i, j)$. Let $g_{ijl}$ denote the fixed cost for accessing mode $l$ between facility $i$ and customer $j$ and $c_{ijl}$ denote the variable cost associated with this mode (see Figure 3.1). Each mode has an associated lower and upper bound denoted by $L_{ij,l-1}$ and $L_{ijl}$, respectively, where $L_{ij0} = 0$. The problem is then to choose which facility sites to open and how to assign customers to the open facilities such that the total costs are minimized.

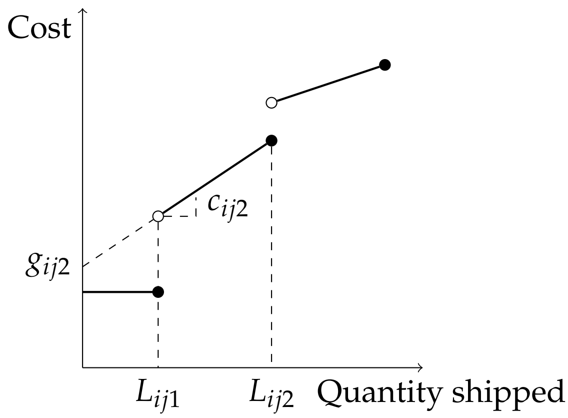



Figure 3.1: A piecewise linear cost function

For both the multiple-choice model and the discretized model we define a binary variable $y_i$ which takes on the value of one if a facility is placed at site $i$ and zero otherwise.

Note that the CFLP with PLTC is $\mathcal{NP}$-hard since it has the CFLP as a special case, which is $\mathcal{NP}$-hard (see e.g. Krarup and Pruzan (1983)). However, we expect the CFLP with PLTC to be much harder to solve in practice for the following reason. The main computational burden in the CFLP is to choose which facility sites to open. Once this has been decided, the remaining problem of assigning customers to the open facilities is easy and can be done by solving a transportation problem by e.g. the network simplex algorithm. Having made the decision of which facilities to open in the CFLP with PLTC, the remaining problem is a network flow problem with piecewise linear costs, which is known to be $\mathcal{NP}$-hard (see e.g. Kim and Pardalos (2000)).

### 3.3.1 The Multiple-Choice Model (MCM)

Define the variable $x_{ijl}$ as the flow on mode $l$ between facility $i$ and customer $j$ and a binary variable, $v_{ijl}$, which is one if the aforementioned mode is used and zero otherwise. Using this notation the capacitated facility location problem with piecewise linear transportation cost can be formulated as

$$
\begin{aligned}
\text{(MCM)} \quad & \min \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{q}(c_{ijl}x_{ijl} + g_{ijl}v_{ijl}) + \sum_{i=1}^{n} f_i y_i \,, && (3.1)\\
& \text{s.t.} \quad \sum_{i=1}^{n}\sum_{l=1}^{q} x_{ijl} = d_j, \qquad \forall\, j, && (3.2)\\
& \sum_{l=1}^{q} v_{ijl} \le 1, \qquad \forall\,(i,j), && (3.3)\\
& \sum_{j=1}^{m}\sum_{l=1}^{q} x_{ijl} \le S_i y_i, \qquad \forall\, i, && (3.4)\\
& x_{ijl} \le L_{ijl} v_{ijl}, \qquad \forall\,(i,j,l), && (3.5)\\
& x_{ijl} \ge L_{ij,l-1} v_{ijl}, \qquad \forall\,(i,j,l), && (3.6)\\
& x_{ijl} \ge 0, \qquad \forall\,(i,j,l), && (3.7)\\
& y_i \in \{0,1\}, \qquad \forall\, i, && (3.8)\\
& v_{ijl} \in \{0,1\}, \qquad \forall\,(i,j,l). && (3.9)
\end{aligned}
$$

Equation (3.1) states that the objective is to minimize the total costs, i.e. the location costs and the transportation costs. The constraint (3.2) ensures that each customer's demand is met. This constraint could be modeled as an inequality, but equality will hold in any optimal solution since the cost function is non-decreasing. Equation (3.3) states that no more than one mode of transportation may be used between each pair of facility and customer. Constraint (3.4) states that the maximum capacity at a facility must not be exceeded. The constraints (3.5) and (3.6), respectively, ensure that the upper and lower bound on each mode are obeyed.

### 3.3.2 The Discretized Model (DM) (proposed by Correia et al. (2010))

Define $\alpha_{ij}^{b}$ as the cost of supplying $b$ units on arc $(i,j)$ where $b = 1,\ldots,L_{ij}$ and $L_{ij} = \min\{S_i, d_j\}$. We define a binary variable $z_{ij}^{b}$ for each arc $(i,j)$ and possible value of $b$, which takes on a value of one if there is a flow of $b$ units on the arc and zero otherwise. The capacitated facility location problem with piecewise linear transportation cost can be formulated as (proposed by Correia et al. (2010))

$$\text{(DM)} \quad \min \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{b=1}^{L_{ij}} \alpha_{ij}^{b} z_{ij}^{b} + \sum_{i=1}^{n} f_i y_i, \tag{3.10}$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\sum_{b=1}^{L_{ij}} b z_{ij}^{b} = d_j, \qquad \forall\, j, \tag{3.11}$$

$$\sum_{j=1}^{m}\sum_{b=1}^{L_{ij}} b z_{ij}^{b} \leq S_i y_i, \qquad \forall\, i, \tag{3.12}$$

$$\sum_{b=1}^{L_{ij}} z_{ij}^{b} \leq 1, \qquad \forall\, (i,j), \tag{3.13}$$

$$y_i \in \{0,1\}, \qquad \forall\, i, \tag{3.14}$$

$$z_{ij}^{b} \in \{0,1\}, \qquad \forall\, (i,j,b). \tag{3.15}$$

The objective, defined by equation (3.10), is to minimize the total costs consisting of the cost of opening the facilities and the costs for servicing the customers. Each customer has to be satisfied by constraint (3.11) and the capacity at the facilities cannot be exceeded by equation (3.12). The constraint (3.11) could be modeled as an inequality. For each arc, at most one flow can be chosen by constraint (3.13). Note, that this formulation is applicable to any separable cost function as long as an optimal integer solution exists.

### 3.3.3 Equivalence of the Linear Programming Relaxation

It can be showed that the two proposed formulations provide identical linear programming bounds. We can therefore place the discretized model in the same category as the models considered by Croxton et al. (2003b) and Vielma et al. (2010) as the linear programming relaxations in these models approximate the piecewise linear cost function by its lower convex envelope.

**Theorem 1**

*The optimal value of the LP relaxation of the DM is the same as that of the MCM.*

The general idea can be found in Duhamel et al. (2012) where a similar proof is proposed for a different kind of network flow problem.

## 3.4 Valid Inequalities

In this section we consider a number of valid inequalities for both the MCM and the DM. A well-known valid inequality for capacitated facility location models is

$$\sum_{i=1}^{n} y_i \geq S^{\min}, \tag{3.16}$$

where

$$S^{\min} = \min \sum_{i=1}^{n} y_i, \tag{3.17}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} S_i y_i \geq D := \sum_{j=1}^{m} d_j, \tag{3.18}$$

$$y_i \in \{0,1\}, \qquad \forall\, i. \tag{3.19}$$

This inequality is directly applicable to both models. Note, that in Correia et al. (2010) the computation of $S^{\min}$ reduces to $S^{\min} = \lceil S/D \rceil$ as the facilities they consider have identical capacities $S = S_i, \forall i$.

### 3.4.1 To the DM (originally by Correia et al. (2010))

The first set of valid inequalities specifically for the DM is based on integer rounding (see e.g. Wolsey (1998)). They are derived from the constraint (3.11) by dividing the equation by an integer $t$ and then rounding the coefficients either up or down. The resulting valid inequalities, proposed by Correia et al. (2010), are

$$\sum_{i=1}^{n}\sum_{b=1}^{L_{ij}}\left\lceil\frac{b}{t}\right\rceil z_{ij}^{b} \geq \left\lceil\frac{d_j}{t}\right\rceil, \qquad \forall\, j,\; t=2,\ldots,d_j, \tag{3.20}$$

$$\sum_{i=1}^{n}\sum_{b=1}^{L_{ij}}\left\lfloor\frac{b}{t}\right\rfloor z_{ij}^{b} \leq \left\lfloor\frac{d_j}{t}\right\rfloor, \qquad \forall\, j,\; t=2,\ldots,d_j. \tag{3.21}$$

Correia et al. also proposes to add the inequality

$$\sum_{b=1}^{L_{ij}} b z_{ij}^{b} \leq \min\{d_j, S_i\} y_i, \qquad \forall\,(i,j). \tag{3.22}$$

In Correia et al. (2010) one additional inequality is proposed, but they conclude that it is in general inefficient as it leads to an increase in the computation time on most instances and therefore we do not consider it here. We also add the following inequality

$$\sum_{b=1}^{L_{ij}} z_{ij}^{b} \leq y_i, \qquad \forall\,(i,j), \tag{3.23}$$

which can be seen to be valid since there can be no flow outbound from a closed facility site. It was not considered in Correia et al. (2010), but is a slightly modified version of the well-known *variable upper bound* constraint often added in facility location models (see e.g. Cornuejols et al. (1991) and Klose and Drexl (2005b)).

### 3.4.2 To the MCM

In this section we propose valid inequalities for the MCM similar to the valid inequalities (3.20)–(3.23) proposed for the discretized model. To derive valid inequalities similar in nature to equations (3.20) and (3.21), we start out by considering the following inequalities, which are redundant in an optimal solution,

$$\sum_{i=1}^{n}\sum_{l=1}^{q} L_{ijl} v_{ijl} \geq d_j, \qquad \forall\, j, \tag{3.24}$$

and

$$\sum_{i=1}^{n}\sum_{l=1}^{q} L_{ij,l-1} v_{ijl} \leq d_j, \qquad \forall\, j. \tag{3.25}$$

The constraint (3.24) ensures that sufficient capacity is installed on each arc to supply each customer. The constraint (3.25) ensures that the minimum flow to a customer is not greater than his demand as this would imply that the solution is suboptimal since the cost function is non-decreasing. Using integer-rounding we obtain the following valid inequalities for the multiple-choice model,

$$\sum_{i=1}^{n}\sum_{l=1}^{q} \left\lceil \frac{L_{ijl}}{t} \right\rceil v_{ijl} \geq \left\lceil \frac{d_j}{t} \right\rceil, \qquad \forall\, j,\ t = 1, \ldots, d_j, \tag{3.26}$$

and

$$\sum_{i=1}^{n}\sum_{l=1}^{q} \left\lfloor \frac{L_{ij,l-1}}{t} \right\rfloor v_{ijl} \leq \left\lfloor \frac{d_j}{t} \right\rfloor, \qquad \forall\, j,\ t = 2, \ldots, d_j. \tag{3.27}$$

We propose the following inequalities, which are similar in nature to equations (3.22) and (3.23),

$$x_{ijl} \leq \min\{d_j, S_i, L_{ijl}\} y_i, \qquad \forall\, (i, j, l), \tag{3.28}$$

$$\sum_{l=1}^{q} v_{ijl} \leq y_i, \qquad \forall\, (i, j). \tag{3.29}$$

## 3.5 A Lagrangean Heuristic

In this section we present a Lagrangean heuristic for the capacitated facility location problem with piecewise linear transportation costs to obtain upper and lower bounds on the problem and potentially reduce the problem size by pegging $y$-variables. In Subsection 3.5.1 we describe how variable pegging can reduce the problem size and in Subsection 3.5.2 we present a number of primal heuristics. The entire Lagrangean heuristic is summarized in Subsection 3.5.3. The remainder of this section outlines the relaxation.

The Lagrangean heuristic is based on the MCM, where we relax the demand constraints (3.2) in a Lagrangean manner with multipliers $\lambda_j$. The relaxed problem can be formulated as a master problem with $n$ subproblems. Each subproblem is defined as

$$\begin{aligned}
\pi_i = \min & \sum_{j=1}^{m} \sum_{l=1}^{q} ((c_{ijl} - \lambda_j) x_{ijl} + g_{ijl} v_{ijl}) \\
\text{s.t. } & \sum_{l=1}^{q} v_{ijl} \leq 1, \qquad \forall j, \\
& \sum_{j=1}^{m} \sum_{l=1}^{q} x_{ijl} \leq S_i, \\
& x_{ijl} \leq L_{ijl} v_{ijl}, \qquad \forall (j,l), \\
& x_{ijl} \geq L_{ij,l-1} v_{ijl}, \qquad \forall (j,l), \\
& x_{ijl} \geq 0, \qquad \forall (j,l), \\
& v_{ijl} \in \{0,1\}, \qquad \forall (j,l),
\end{aligned}$$

which is a *Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem* (SSFCMCTP). It is solved by a specialized dynamic programming algorithm, for which details can be found in Christensen et al. (2012). Using this we can rewrite the Lagrangean relaxation as

$$Z_{\text{LR}}(\lambda^t) = \sum_{j=1}^{m} d_j \lambda_j^t + \min \sum_{i=1}^{n} (f_i + \pi_i) y_i, \tag{3.30}$$

$$\text{s.t. } y_i \in \{0,1\}, \qquad \forall i. \tag{3.31}$$

The program (3.30)–(3.31) can be used directly to obtain lower bounds, but it is not well suited as a basis for primal heuristics for two reasons. Firstly, a solution might not provide sufficient facility capacity to cover the total demand. Secondly, it might not provide enough capacity on the modes to fully supply each customer. In order to resolve this, we add the inequality (3.16), as well as the following

$$\sum_{i=1}^{n} y_i L_{ijq} \geq d_j, \qquad \forall j, \tag{3.32}$$

which can be derived by combining equations (3.24) and (3.29). Besides helping us to recover a primal solution, this also strengthens the lower bound obtained, which in turn makes it easier to reduce the problem size by variable pegging. Thus, the master problem used is min{(3.30) s.t. (3.16), (3.31), (3.32)}. This is a *d-dimensional Knapsack Problem* ($d$-KP) and is $\mathcal{NP}$-hard (see e.g. Kellerer et al. (2003)). In our implementation the $d$-KP was solved by passing the problem to a general MIP solver (CPLEX version 12.4) and for the test instances used in this paper the $d$-KP could be solved sufficiently fast. In order to find the multipliers that yields the best lower bound, we use the popular subgradient optimization described by Held et al. (1974). If $\lambda^t$ is the vector of multipliers in iteration $t$, we update these according to

$$\lambda^{t+1} = \lambda^t + \theta_t r^t, \tag{3.33}$$

where $\theta_t$ is the step length and $r^t$ is the current subgradient. The step length is found as

$$\theta^t = \beta_t (z_{\text{UB}} - z_{LR}(\lambda^t)) / \|r^t\|^2, \tag{3.34}$$

where $\beta_t$ is a step length parameter initially set to 2 and $z_{\text{UB}}$ is the best upper bound as found by the heuristics in Subsection 3.5.2. The step length is halved if no improvement in $z_{\text{UB}}$ was found in the last 5 iterations. The subgradient's $j$th element is given by

$$r_j^t = d_j - \sum_{\{i:y_i=1\}} \sum_{l=1}^{q} x_{ijl}^{\pi*}, \tag{3.35}$$

where $x_{ijl}^{\pi*}$ is the optimal solution to the corresponding subproblem $\pi_i$.

### 3.5.1 Variable Pegging

Variable pegging is a technique to fix a variable, usually a binary variable, to a specific value and then compute a lower bound on the resulting objective function. This is often employed in the facility location literature where one tries to peg the $y$-variables to either zero or one (see e.g. Beasley (1993) and Sridharan (1995)). This is done by considering the optimal solution, $(y_i^*)_{i=1}^n$ to the $d$-KP in a given iteration and for each variable $y_i$ considering the $d$-KP resulting when fixing the value of $y_i$ to $1 - y_i^*$. If the optimal solution to this restricted problem yields an objective function value higher than the current best, $z_{\mathrm{UB}}$, then the variable can be fixed to $y_i^*$ in any improving solution. Since this requires solving a $d$-KP each time we try to peg a $y$-variable, we only employ this every 10 iterations.

### 3.5.2 Primal Heuristics

From the Lagrangean relaxation we obtain lower bounds on the problem, as well as a suggestion on where to locate facilities through the solution to the $d$-KP. Here we propose a number of primal heuristics exploiting the information from the master problem by only considering the facilities that are open in the optimal solution to the current $d$-KP. If one of these heuristics finds a feasible solution then we fix the modes used in the solution and solve the corresponding transportation problem. This ensures that the optimal solution will be recovered if the optimal combination of modes is used, just like solving a transportation problem given the optimal placement of facilities in the CFLP will yield the optimal solution. The following primal heuristics are employed:

#### Linearized Cost (proposed by Kim and Pardalos (2000))

This heuristic linearizes the cost function and iteratively updates the linearized cost coefficient according to the optimal solution of the corresponding transportation problem. This is shown in Figure 3.2, where $\bar{c}_{ij}^t$ is the linearized cost used in iteration $t$ between facility $i$ and customer $j$ and $\sum_{l=1}^{q} x_{ijl}^{\pi *}$ is the flow on the arc in the optimal solution in iteration $t$. In each iteration the optimal solution is used to obtain a primal solution. This update procedure stops when no cost coefficient needs to be updated or when 5 iterations have been made.

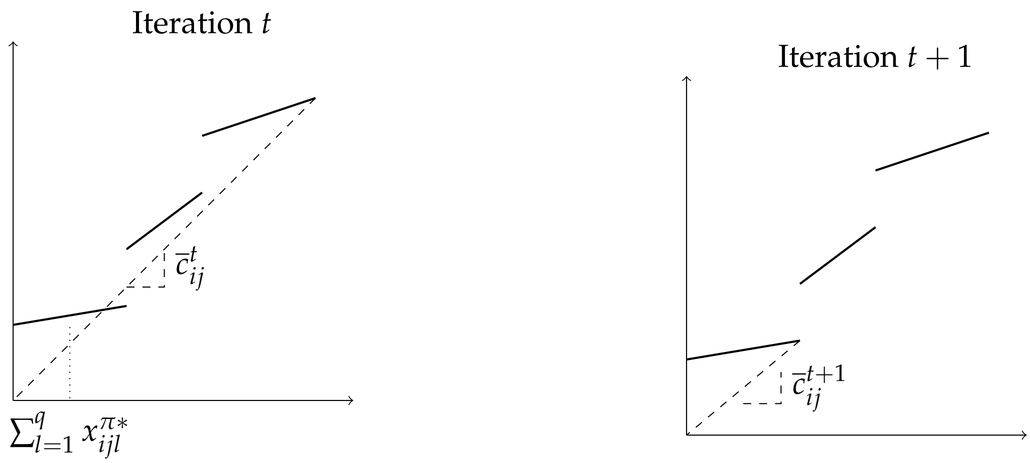



Figure 3.2: General principle of updating the linearized cost as in Kim and Pardalos (2000)

**Least Cost Addition**

This heuristic aims to construct a solution based on the corresponding optimal solution of the SSFCMCTP associated with each open facility. This is done in a way that ensures that no facility is assigned more demand than its capacity and that no customer gets served more than his demand. If a customer is not fully served, we try to assign the missing demand to the facility having the smallest cost of doing so. When a new solution to $d$-KP is found, the heuristic is run once without using the information from the solution to the SSFCMCTPs.

**Largest Cost Saving**

This heuristic starts out by setting the supply from all the open facilities to the customers to the maximal amount, i.e. $L_{ijq}$. Then it generates a list of customers with an excess supply and sorts the list in non-increasing order. We start with he customer with the highest excess and for each open facility we calculate the savings obtained by dropping this customer (either completely or until the demand is exactly met). In the end the facility with the highest saving is unassigned the customer. This heuristic is also used in a variant in which it will not set supply below the value of the flow from the optimal solution to the SSFCMCTP.

**Candidate Segment**

This heuristic is very similar to the "least cost addition" heuristic, except that it does not consider the cost, but instead focuses solely on obtaining a feasible solution. It starts out by trying to set the flow as in the optimal solution to the SSFCMCTPs without violating neither the capacity nor the demand constraint. Any customer not satisfied is then assigned to an open facility with spare capacity.

### 3.5.3 The Heuristic Step-by-step

To summarize, the steps of the Lagrangean heuristic are

**Step 1)** Solve the SSFCMCTP's by dynamic programming.

**Step 2)** Solve the corresponding $d$-KP and update the best lower bound, $LB$, if it improves the current best lower bound. Additionally, for every 10 iterations run the variable pegging.

**Step 3)** Run the primal heuristics using the optimal solution to the current $d$-KP. If an improvement is found, then store the solution in $UB$.

**Step 4)** If the gap $100 \cdot (UB - LB)/LB$ is smaller than 1%, or the maximum number of iterations has been performed, then stop, otherwise update the Lagrangean multipliers and go to Step 1.

In the computational experiments conducted in this paper we limited the heuristic to at most 50 iterations.

## 3.6 Computational Results

In this section we compare the two formulations on a test bed of randomly generated instances. We compare the effect of adding the valid inequalities proposed to each model on the lower bound and the total computation time required to solve the instances to optimality. We also test the bounds provided by the Lagrangean heuristic as well as the effect of initializing both formulations with information obtained from the heuristic, i.e. the upper bound and the facility location variables fixed to either zero or one.

### 3.6.1 Test Instances

The test instances were generated as follows. The instances have either 10 or 20 potential facility sites and the number of customers are either 50, 100 or 200. We consider instances with either 3, 5 or 7 modes. The fixed cost for operating a facility site was drawn from $UI[2000, 5000]$, where $UI[x, y]$ denotes the discrete uniform distribution over the interval $[x, y]$. The demand was drawn from $UI[20, 60]$ and the facility capacity from $UI[600, 1000]$ except for the instances with 10 facility sites and 200 customers where the capacity was drawn from $UI[1200, 2000]$ in order to ensure feasibility. The capacities of each mode were set to $L_{ijl} = \lfloor l \cdot d_j / q \rfloor$. The fixed cost for the first mode, $g_{ij1}$, was drawn from $UI[10, 40]$ and the subsequent fixed cost for modes was calculated as $g_{ijl} = 1.2 \cdot g_{ij,l-1}$. The variable cost on the first mode was drawn from $UI[1, 10]$ and the variable cost for the subsequent modes was set to $c_{ijl} = 0.8 \cdot c_{ij,l-1}$ such that higher modes have increased fixed costs, but a smaller per unit cost part. We generate five instances for each configuration considered in Subsection 3.6.2.

### 3.6.2 Results

All tests were conducted on a laptop with 4 GB RAM and an Intel Core 2 Duo 2.4 GHz processor running Linux. The two formulations and the Lagrangean heuristic were implemented in C++ and interfaced with CPLEX (version 12.4) using the Concert technology.

Table 3.1 presents the strength of the linear programming relaxation of both the MCM and the DM with and without the relevant valid inequalities proposed. Additionally, the strength of the lower bound provided by the Lagrangean heuristic is displayed. For each of these methods we display the results as $\alpha; \beta$, where $\alpha$ is the gap provided by the lower bound obtained by the algorithm and the optimal solution in percentage (or the best known solution if the optimal solution is not known). $\beta$ denotes the runtime in CPU seconds to obtain the lower bound.

Table 3.1: Tightness of the linear programming relaxation for the MCM and the DM both w. and w.o. the valid inequalities and the lower bound provided by the Lagrangean heuristic.

| $m, q$ | MCM | MCM+(3.16)+ (3.26)–(3.29) | DM | DM+(3.16)–(3.23) | Lag. Heur. |
|---|---|---|---|---|---|
| 10 facility instances | | | | | |
| 50, 3 | 19.06; 0.02 | 0.11; 0.43 | 19.06; 0.07 | 0.11; 1.24 | 0.08; 0.68 |
| 50, 5 | 16.30; 0.03 | 1.09; 1.06 | 16.30; 0.08 | 1.09; 1.08 | 0.07; 0.80 |
| 50, 7 | 22.21; 0.03 | 1.30; 4.18 | 22.21; 0.07 | 1.30; 1.78 | 1.09; 1.31 |
| 100, 3 | 4.87; 0.03 | 0.11; 1.01 | 4.87; 0.20 | 0.11; 2.50 | 0.06; 1.56 |
| 100, 5 | 8.14; 0.06 | 1.83; 3.99 | 8.14; 0.29 | 1.83; 4.03 | 0.38; 2.40 |
| 100, 7 | 7.55; 0.10 | 2.28; 9.13 | 7.55; 0.29 | 2.35; 4.36 | 0.79; 2.51 |
| 200, 3 | 9.98; 0.08 | 0.46; 5.88 | 9.98; 0.47 | 0.46; 8.15 | 0.30; 5.07 |
| 200, 5 | 7.81; 0.13 | 0.30; 18.04 | 7.81; 0.41 | 0.30; 8.64 | 0.13; 4.70 |
| 200, 7 | 8.35; 0.27 | 0.55; 45.72 | 8.35; 0.48 | 0.54; 9.71 | 1.13; 4.99 |
| 20 facility instances | | | | | |
| 50, 3 | 25.95; 0.03 | 1.77; 2.23 | 25.95; 0.24 | 1.77; 4.27 | 0.43; 1.53 |
| 50, 5 | 28.89; 0.07 | 1.60; 4.58 | 28.89; 0.20 | 1.60; 3.84 | 0.10; 1.11 |
| 50, 7 | 27.95; 0.09 | 3.62; 14.76 | 27.95; 0.24 | 3.61; 4.98 | 1.76; 2.02 |
| 100, 3 | 10.20; 0.09 | 0.32; 4.00 | 10.20; 0.56 | 0.32; 7.54 | 0.16; 3.13 |
| 100, 5 | 9.01; 0.16 | 1.32; 22.52 | 9.01; 0.53 | 1.32; 11.34 | 0.51; 4.23 |
| 100, 7 | 8.15; 0.19 | 0.13; 41.06 | 8.15; 0.66 | 0.12; 8.19 | 0.62; 3.88 |
| 200, 3 | 3.50; 0.17 | 0.81; 13.21 | 3.50; 1.39 | 0.81; 21.89 | 0.28; 9.65 |
| 200, 5 | 3.30; 0.30 | 1.12; 42.11 | 3.30; 1.39 | 1.12; 29.02 | 0.22; 11.93 |
| 200, 7 | 4.49; 0.63 | 1.48; 121.83 | 4.49; 1.72 | 1.47; 36.37 | 0.91; 12.41 |

For each algorithm we display $\alpha; \beta$ where $\alpha$ is the gap between the lower bound provided by the algorithm and the optimal solution in percentage (or best known solution if optimality was not proven). $\beta$ is the CPU time in seconds to achieve the lower bound.

As expected the gap decreases for both the MCM and the DM when valid inequalities are added at the expense of an increase in computation time. In the tests where the valid inequalities were added we observe that the gap seems to increase as the number of modes increases for both models. The difference in strength for the MCM and the DM after adding the valid inequalities is virtually non-existing with respect to the gap, while the computation time to obtain the bound seems to be slightly lower for the DM on the larger instances. The lower bound provided by the Lagrangean heuristic is almost always better than the bound obtained from either formulations even with the added valid inequalities and very often faster to obtain.

In Table 3.2 we present the time used for solving the instances to optimality. The result is presented for both models with and without valid inequalities and using the information obtained by the Lagrangean heuristic. The latter is done by initializing the model with the best primal solution found by the heuristic and fixing the variables identified by the procedure in Section 3.5.1. For each of these settings we report $\zeta; \eta$ where $\zeta$ is the total computation time in CPU seconds including the time spent running the Lagrangean heuristic, where relevant. $\eta$ is the number of instances that could not be solved either because the algorithm ran out of memory or because it reached the time limit of 10,800 CPU seconds (3 hours). If an instance could not be solved to optimality it is excluded from the average computation time. For the Lagrangean heuristic we display the number of the facility location variables that could be pegged to either zero or one on average.
From Table 3.2 we make the following observations:

i. No model is significantly better than the others. One example of this can be found in the 10 facility instance with 100 customers and 5 modes where the MCM is roughly two times faster than the DM, while the opposite is true in the same setting with 7 modes instead.

ii. Adding the valid inequalities for either model makes it possible for the solver to prove optimality for problems that could otherwise not be solved. However, on some instances this will lead to an increase in the total solution time.

iii. Warm starting CPLEX with the solution from the Lagrangean heuristic can be very effective and decreases the solution time by up to 80% when using the MCM compared

Table 3.2: Time spent to solve the test bed to optimality for the MCM and the DM both with and without the valid inequalities and by warm starting them with the information obtained from the Lagrangean heuristic.

| | MCM | | | DM | | | LH |
|---|---|---|---|---|---|---|---|
| $m, q$ | MCM | MCM+(3.16)+ (3.26)–(3.29) | MCM+(3.16)+ (3.26)– (3.29)+LH | DM | DM+(3.16)– (3.23) | DM+(3.16)– (3.23)+ LH | Avg. No. Vars. Fixed |
| 10 facility instances | | | | | | | |
| 50, 3 | 4.81; 0 | 2.75; 0 | 1.42; 0 | 48.91; 0 | 10.01; 0 | 4.52; 0 | 9.0 |
| 50, 5 | 6.93; 0 | 7.45; 0 | 1.59; 0 | 42.03; 0 | 40.25; 0 | 6.06; 0 | 8.8 |
| 50, 7 | 50.26; 0 | 29.04; 0 | 10.05; 0 | 126.86; 0 | 53.73; 0 | 37.53; 0 | 4.2 |
| 100, 3 | 56.97; 0 | 79.91; 0 | 24.62; 0 | 912.53; 0 | 99.81; 0 | 37.47; 0 | 8.4 |
| 100, 5 | 483.76; 0 | 565.69; 0 | 544.09; 0 | 773.90; 0 | 1206.73; 0 | 1162.41; 0 | 6.0 |
| 100, 7 | 727.21; 3 | 732.53; 0 | 549.65; 0 | 473.60; 0 | 458.77; 0 | 238.53; 0 | 4.6 |
| 200, 3 | 244.95; 0 | 567.68; 0 | 343.43; 0 | 2807.75; 0 | 707.85; 0 | 345.48; 0 | 5.2 |
| 200, 5 | 249.26; 0 | 241.12; 0 | 141.51; 0 | 794.20; 0 | 178.80; 0 | 104.97; 0 | 4.6 |
| 200, 7 | 2022.40; 1 | 402.54; 1 | 103.00; 1 | 3071.37; 1 | 1169.08; 0 | 933.98; 0 | 2.4 |
| 20 facility instances | | | | | | | |
| 50, 3 | 15.80; 0 | 19.43; 0 | 4.04; 0 | 126.41; 0 | 192.29; 0 | 21.43; 0 | 15.6 |
| 50, 5 | 42.48; 0 | 52.59; 0 | 3.69; 0 | 236.25; 0 | 485.65; 0 | 10.72; 0 | 16.4 |
| 50, 7 | 370.86; 0 | 1217.79; 0 | 304.22; 0 | 383.63; 0 | 815.19; 0 | 536.25; 0 | 11.6 |
| 100, 3 | 32.48; 0 | 37.98; 0 | 11.11; 0 | 577.80; 0 | 82.68; 0 | 26.79; 0 | 15.8 |
| 100, 5 | 491.77; 0 | 1788.88; 0 | 231.03; 0 | 1203.84; 0 | 713.14; 0 | 555.09; 0 | 12.6 |
| 100, 7 | 1315.91; 4 | 1207.40; 0 | 388.71; 0 | 1939.85; 0 | 240.37; 0 | 199.50; 0 | 13.4 |
| 200, 3 | 1090.24; 4 | 186.24; 4 | 66.26; 4 | –; 5 | 167.35; 4 | 125.88; 4 | 10.8 |
| 200, 5 | –; 5 | –; 5 | –; 5 | –; 5 | –; 5 | –; 5 | 8.2 |
| 200, 7 | –; 5 | –; 5 | –; 5 | –; 5 | –; 5 | –; 5 | 4.8 |

For each of these settings we report $\zeta; \eta$ where $\zeta$ is the total computation time in CPU seconds including the time spent running the Lagrangean heuristic, where relevant. $\eta$ is the number of instances that could not be solved either because the algorithm ran out of memory or because it reached the time limit of 10,800 CPU seconds (3 hours).

to just adding the valid inequalities. This effect is slightly less pronounced for the DM but a decrease of 50% is still achieved on several instances.

iv. The combined effect of adding the valid inequalities and warm starting the solver for the MCM is a decrease of up to 70% and meant that a total of 7 previously unsolved instances could now be solved to optimality. For the DM the effect is a reduction of up to 89% and one additional instance could be solved to optimality.

In Table 3.3 we investigate the effect of simultaneously increasing the demand and the facility capacity. This is done by changing the distribution of the demand and capacity as shown in the leftmost column in the table. As the discretized model has a variable associated with each value of $b = 1, \ldots, \min\{S_i, d_j\}$ on the arc $(i, j)$ it is very sensitive to this kind of change. Only a few instances cannot be solved to optimality using the MCM (after adding the valid inequalities), but the DM frequently fails to reach the optimal solution. This leads us to conclude that for instances with these characteristics the MCM seems superior. Note, that initializing the DM with the information from the Lagrangean heuristic means that an additional 4 instances could be solved to optimality.

Finally, in Table 3.4 we test each model as a heuristic, i.e. we run it for 60 CPU seconds and check the quality of the primal bound obtained. The Lagrangean heuristic never runs for more than 18 CPU seconds on any instance, due to the maximum number of iterations being reached. For each setting we report $\gamma, \eta$, where $\gamma$ is the gap between the primal solution obtained and the best known solution and $\eta$ is the number of instances for which no primal solution could be found. From Table 3.4 we see that the valid inequalities generally does not improve the solver's ability to find good solutions fast. However, the Lagrangean heuristic produces very good solutions, within 1% of optimality, and clearly outperforms the other candidates on almost all instances.

Table 3.3: Time spent to solve the instances with increased demand and capacity to optimality for the MCM and the DM both with and without the valid inequalities and by warm starting them with the information obtained from the Lagrangean heuristic.

| Demand/Capacity | MCM | | | DM | | | LH |
|---|---|---|---|---|---|---|---|
| | MCM | MCM+ (3.16)+ (3.26)– (3.29) | MCM+ (3.16)+ (3.26)– (3.29)+LH | DM | DM+ (3.16)– (3.23) | DM+ (3.16)– (3.23)+ LH | Avg. No. Vars. Fixed |
| 10 facilities, 100 customers and 7 modes | | | | | | | |
| $UI[20,60]/UI[600,1000]$ | 727.21; 3 | 732.53; 0 | 549.65; 0 | 473.60; 0 | 458.77; 0 | 238.53; 0 | 4.6 |
| $UI[40,80/UI[800,1200]$ | 1565.98; 2 | 1014.88; 1 | 571.05; 1 | 1483.41; 3 | 533.99; 1 | 753.33; 1 | 4.4 |
| $UI[60,100]/UI[1000,1400]$ | 230.93; 3 | 1122.23; 0 | 2270.90; 0 | 2229.47; 2 | 1930.56; 0 | 2103.49; 0 | 5.6 |
| $UI[80,120]/UI[1200,1600]$ | 1232.03; 3 | 4131.96; 1 | 3734.70; 1 | –; 5 | 3008.06; 4 | 8630.52; 2 | 4.6 |
| $UI[100,140]/UI[1400,1800]$ | –; 5 | 1103.87; 0 | 1066.07; 0 | –; 5 | –; 5 | 1586.68; 3 | 3.4 |
| $UI[120,160]/UI[1600,2000]$ | 3751.24; 4 | 1768.87; 1 | 966.89; 1 | 3690.26; 4 | –; 5 | –; 5 | 4.0 |

For each of these settings we report $\zeta;\eta$ where $\zeta$ is the total computation time in CPU seconds including the time spent running the Lagrangean heuristic, where relevant. $\eta$ is the number of instances that could not be solved either because the algorithm ran out of memory or because it reached the time limit of 10,800 CPU seconds (3 hours).

## 3.7 Conclusion

In this paper we considered a capacitated facility location problem with a piecewise linear, non-decreasing transportation cost function as such functions arise in many real-world applications. We compared two formulations of the problem, namely the multiple-choice model and a discretized model. We reviewed a number of valid inequalities, known as well as new, for both formulations and tested the effect of these inequalities on a number of test instances. Additionally, we proposed a Lagrangean heuristic to further speed-up the solution process. Considering both the valid inequalities and the Lagrangean heuristic we were able to reduce the solution time by up to 80% and to solve a number of instances that could not be solved without these speed-up techniques.

Table 3.4: The optimality gap resulting after running the algorithm for 60 CPU seconds.

| $m, q$ | MCM | | DM | | LH |
|---|---|---|---|---|---|
| | MCM | MCM+(3.16)+ (3.26)–(3.29) | DM | DM+(3.16)– (3.23) | |
| **10 facility instances** | | | | | |
| 50, 3 | 0.00; 0 | 0.00; 0 | 0.01; 0 | 0.00; 0 | 0.07; 0 |
| 50, 5 | 0.00; 0 | 0.00; 0 | 0.04; 0 | 0.00; 0 | 0.04; 0 |
| 50, 7 | 0.14; 0 | 0.07; 0 | 2.99; 0 | 0.07; 0 | 0.01; 0 |
| 100, 3 | 0.00; 0 | 0.01; 0 | 0.33; 0 | 0.23; 0 | 0.15; 0 |
| 100, 5 | 0.26; 0 | 0.41; 0 | 2.46; 0 | 1.31; 0 | 0.06; 0 |
| 100, 7 | 3.01; 0 | 0.56; 0 | 6.17; 0 | 2.13; 0 | 0.09; 0 |
| 200, 3 | 0.02; 0 | 0.20; 0 | 22.60; 0 | 1.56; 2 | 0.05; 0 |
| 200, 5 | 0.04; 0 | 30.23; 0 | 20.40; 0 | 9.33; 2 | 0.13; 0 |
| 200, 7 | 17.24; 0 | 29.10; 0 | 27.08; 0 | 0.00; 2 | 0.02; 0 |
| **20 facility instances** | | | | | |
| 50, 3 | 0.00; 0 | 0.00; 0 | 4.26; 0 | 1.34; 0 | 0.00; 0 |
| 50, 5 | 0.00; 0 | 0.00; 0 | 1.21; 0 | 0.00; 0 | 0.00; 0 |
| 50, 7 | 2.69; 0 | 3.66; 0 | 15.85; 0 | 0.89; 0 | 0.05; 0 |
| 100, 3 | 0.00; 0 | 0.00; 0 | 39.46; 0 | 3.43; 0 | 0.11; 0 |
| 100, 5 | 1.61; 0 | 4.50; 0 | 50.90; 0 | 4.70; 0 | 0.08; 0 |
| 100, 7 | 13.62; 0 | 4.21; 0 | 23.67; 0 | 9.11; 0 | 0.06; 0 |
| 200, 3 | 0.80; 0 | 38.00; 0 | 31.98; 0 | 18.11; 2 | 0.23; 0 |
| 200, 5 | 2.29; 0 | 49.76; 0 | 41.97; 0 | –; 5 | 0.24; 0 |
| 200, 7 | 88.75; 2 | 181.35; 1 | 41.94; 1 | 18.11; 4 | 0.09; 0 |

For each setting we report $\gamma; \eta$, where $\gamma$ is the gap between the primal solution obtained and the best known solution and $\eta$ is the number of instances for which no primal solution could be found.

## 3.A The Capacitated Facility Location Problem with Modular Transportation Cost

Here we show how to transform an instance of the capacitated facility location problem with modular transportation cost to an instance of the capacitated facility location problem with piecewise linear transportation cost.

We define $I, J, S_i, d_j$ and $f_i$ as in Section 3.3. Besides locating facilities, we also have to choose a combination of modules that ensures sufficient capacity is installed on each arc in use at minimum costs. Let $K = \{1, \ldots, w\}$ denote the set of modules and $h_k$ the fixed cost for using one module of type $k$ with an associated capacity of $p_k$. Additionally, we incur a variable cost per flow unit on arc $(i,j)$ of $c_{ij}$. The capacitated facility location problem with modular transportation cost involves deciding on which facilities to open and how to install capacity on the arcs serving customers such that the total costs comprising the fixed operating costs, variable transportation costs and costs for installing modules on the arcs are minimized.

### 3.A.1 Problem Transformation

As the problem of choosing a cost minimizing combination of modules is independent of the arc considered we can decouple this from the rest of the model and solve it separately. This is the key observation when reformulating the problem. The problem of choosing a cost minimizing set of modules that ensures a capacity of at least $b$ units can be formulated as a knapsack problem,

$$\text{KP}(b) \quad \alpha_b = \min \sum_{k=1}^{w} h_k \mu_k(b), \tag{3.36}$$

$$\text{s.t.} \sum_{k=1}^{w} p_k \mu_k(b) \geq b, \tag{3.37}$$

$$\mu_k(b) \geq 0 \text{ and integer }, \qquad \forall k. \tag{3.38}$$

We denote the optimal solution to the above problem $(\mu_k^*(b))_{k=1}^{w}$ and define $L_{ij}$ as the largest flow relevant on the arc between $i$ and $j$, i.e. $L_{ij} = \min\{S_i, d_j\}$. We solve the problem KP($b$) for each possible flow, i.e. for $b = 1, \ldots, \max_{(i,j)}\{L_{ij}\}$. Note, that if the optimal solution to KP($b$) has an installed capacity higher than $b$, i.e. $\sum_{k=1}^{w} p_k \mu_k^*(b) > b$, we may proceed directly to the program where $b' = \sum_{k=1}^{w} p_k \mu_k^*(b) + 1$ and thereby skip solving a number of problems. It is possible to use specialized algorithms that can also exploit the optimal solution to the program KP($b$) when solving the program KP($b+1$) (see e.g. Andonov et al. (2000)).

The following procedure transforms an instance with modular transportation cost to an instance with a piecewise linear cost structure.

**Step 1)** Select an unprocessed arc $(i, j)$. If none exists, then stop.

**Step 2)** Initialize by setting $b = 1$ and $PreviousValue = 0$.

**Step 3)** Obtain the optimal solution to KP$(b)$.

**Step 4)** Let $\overline{b} = \min\{d_j, S_i, \sum_{k=1}^{w} p_k \mu_k^*(b)\}$. Construct a new mode $l$ and set $L_{ijl} = \overline{b}$, $g_{ijl} = \alpha_b$, $c_{ijl} = c_{ij}$ and $L_{ij,l-1} = PreviousValue$.

**Step 5)** Set $PreviousValue = \overline{b}$ and $b = \overline{b} + 1$.

**Step 6)** If $b \leq \min\{d_j, S_i\}$, go to Step 3. Otherwise we have fully processed this arc and go to Step 1.

When we have a solution to an instance with piecewise linear transportation costs then the corresponding solution to the original instance with modular transportation costs can be found by backtracking. Note, that this procedure can also be used to convert an instance of the capacitated facility location problem with *machine siting* (see e.g. ReVelle and LaPorte (1996) and Cañavate-Bernal et al. (2000)) into an instance of the capacitated facility location problem with piecewise linear production costs.

# Chapter 4

# A Branch-Cut-and-Price Algorithm for the Piecewise Linear Transportation Problem

# A Branch-Cut-and-Price algorithm for the Piecewise Linear Transportation Problem

Tue R. L. Christensen† and Martine Labbé*

†Aarhus University, Department of Economics and Business, Denmark

tuec@asb.dk

*Université Libre de Bruxelles, Département d'Informatique, Belgium, mlabbe@ulb.ac.be

**Abstract**

In this paper we present an exact solution method for the transportation problem with piecewise linear costs. This problem is fundamental within supply chain management and is a straightforward extension of the *fixed-charge transportation problem*. We consider two Dantzig-Wolfe reformulations and investigate their relative strength with respect to the linear programming (LP) relaxation both theoretical and practical through tests on a number of instances. Based on one of the proposed formulations we derive an exact method by branching and adding generalized upper bound constraints from violated cover inequalities. The proposed solution method is tested on a set of randomly generated instances and compares favorably to solving the model using a standard formulation solved by a state-of-the-art commercial solver.

## 4.1 Introduction

In this paper we consider the problem of finding a minimum cost flow in a bipartite graph between a set of suppliers and a set of customers. The cost of sending goods on an arc follows a piecewise linear structure (see Figure 4.1) and the problem is thereby a natural generalization of the *Fixed-Charge Transportation Problem*. This problem is termed the *Piecewise Linear Transportation Problem* (PLTP) and is a versatile problem that is fundamental within supply chain network design and arises in a number of applications. The general form of the cost functions allows for modeling of different transportation modes such as *small packages, less-than-truckloads, truckloads,* and *air freight* (see e.g. Croxton et al. (2003b); Lapierre et al. (2004)). Additionally, the cost function can be used to model price discounts such as all-unit or incremental discounts, often found in procurement theory (see Davenport and Kalagnanam (2001), Kameshwaran and Narahari (2009)) or to linearize an otherwise non-linear cost function.

Kim and Pardalos (2000) presents a heuristic for the PLTP based on a linearization of the cost function and subsequent solution of a (standard) transportation problem. In Croxton et al. (2003a) the authors show that the linear programming relaxations of three textbook formulations of a piecewise linear function are equivalent. One of these three is the *Multiple-Choice Model* (MCM) used in Subsection 4.2.1. Other studies (e.g. Keha et al. (2004); Vielma et al. (2008, 2010)) have extended this result to include a number of other formulations and they also perform tests to find the best formulation in terms of solving the problem to optimality by a standard solver. The most recent of these studies suggests that when the number of modes is relatively small (as in our tests), the MCM, presented in Subsection 4.2.1, is preferable. As the linear programming relaxation of the standard models is often very poor, we propose two stronger formulations for the problem, both based on a Dantzig-Wolfe reformulation of the problem.

In Section 4.2 we give a formal definition of the problem using the standard multiple-choice formulation and two new formulations. The two new formulations rely on a Dantzig-

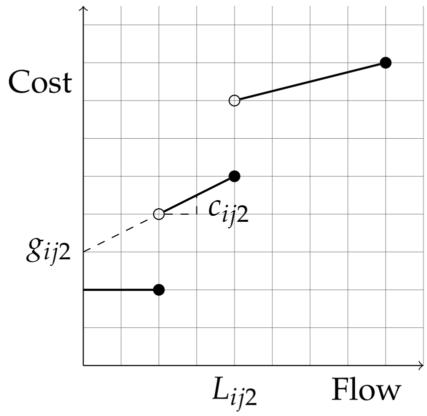



Figure 4.1: A piecewise linear cost function

Wolfe reformulation of the original problem and column generation is required to solve the LP relaxation. The theoretical strength of the formulations is investigated in Section 4.3, along with the actual strength on a test bed of instances. Based on these results we extend one formulation into an exact solution method by adding the valid inequalities described in Section 4.4 and by applying the branching rule described in Section 4.5. In Section 4.6 we test the solution method on a number of randomly generated instances and compare the method to solving the MCM by a standard commercial solver. Section 4.7 summarizes our findings and concludes this paper.

## 4.2 Mathematical Formulations

In this section we first define the PLTP and introduce selected notations. Then we introduce a standard formulation and two new formulations of the PLTP.

Let the set of supply nodes (suppliers) be denoted by the set $I = \{1, \ldots, n\}$. The total capacity of each supplier $i$ is denoted by $S_i$. The demand nodes (customers) are denoted by the set $J = \{1, \ldots, m\}$, where customer $j$ has demand $d_j$. The cost of transporting goods from supplier $i$ to customer $j$ follows a piecewise linear cost structure with $q_{ij}$ line segments on the arc between supplier $i$ and customer $j$, which is also known as the modes. For notational convenience we will assume that $q_{ij} = q$ for all $(i,j)$ and we denote the set of modes by $Q$. Each mode $l \in Q$ from $i$ to $j$ is characterized by a fixed cost for using the mode, $g_{ijl}$ and a variable cost (the slope of the mode), $c_{ijl}$. Additionally, the flow using mode $l$ on the arc $(i,j)$ is restricted to a minimum of $L_{ij,l-1}$ and a maximum of $L_{ijl}$ (see Figure 4.1), where $L_{ij0} = 0$. We assume that $L_{ijq} \leq \min\{S_i, d_j\}$, i.e. that the maximum flow between supplier $i$ and customer $j$ does not exceed neither the capacity nor the demand, respectively. Note, that the maximum capacity on an arc might be restrictive, i.e. the inequality might be strict.

### 4.2.1 The Multiple-Choice Model

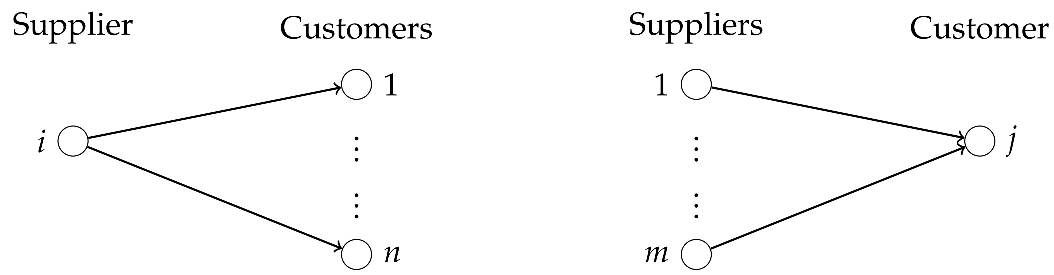
One standard way of representing piecewise linear functions is by the so-called *Multiple-Choice Model*. Using this formulation, the problem can be stated as

$$
\begin{aligned}
\text{(MCM)} \quad \min & \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{q}(c_{ijl}x_{ijl} + g_{ijl}v_{ijl}), && && (4.1)\\
\text{s.t.} & \sum_{i=1}^{n}\sum_{l=1}^{q} x_{ijl} = d_j, && \forall j \in J, && (4.2)\\
& \sum_{l=1}^{q} v_{ijl} \leq 1, && \forall (i,j), i \in I, j \in J && (4.3)\\
& \sum_{j=1}^{m}\sum_{l=1}^{q} x_{ijl} \leq S_i, && \forall\, i \in I, && (4.4)\\
& x_{ijl} \leq L_{ijl}v_{ijl}, && \forall (i,j,l), i \in I, j \in J, l \in Q && (4.5)\\
& x_{ijl} \geq L_{ij,l-1}v_{ijl}, && \forall (i,j,l), i \in I, j \in J, l \in Q && (4.6)\\
& x_{ijl} \geq 0, && \forall (i,j,l), i \in I, j \in J, l \in Q && (4.7)\\
& v_{ijl} \in \{0,1\}, && \forall (i,j,l), i \in I, j \in J, l \in Q. && (4.8)
\end{aligned}
$$

The objective, (4.1), is to minimize the total costs of supplying the customers. Each customer has to receive an amount equal to his demand by (4.2). Constraints (4.3) enforce that at most one mode is used between each combination of supplier and customer. Constraints (4.4) state that the solution has to obey the capacity at each supplier. Constraints (4.5) and

(4.6) bound the flow within the associated upper and lower bounds for mode $l$ between the supplier $i$ and customer $j$, respectively. Variable $x_{ijl}$ represents the flow between $i$ and $j$ on the mode $l$ and the associated binary variable $v_{ijl}$ is one if this mode is used, zero otherwise. We known that the linear programming relaxation of the MCM is equivalent to replacing the objective function by its lower convex envelope (see e.g. Croxton et al. (2003a)). We denote by MCM-LP the relaxed problem defined by the program (4.1)–(4.8), where (4.8) is replaced by the constraints $0 \leq v_{ijl} \leq 1, \forall(i,j,l), i \in I, j \in J, l \in Q$.

The two new formulations are based on the introduction of variables for each feasible flow-vector from either each supplier to all the customers, or from all suppliers to each customer (see Figure 4.2). These models are denoted the *Supplier-Based Model* (SBM) and the *Customer-Based Model* (CBM), respectively. Both formulations are obtained by applying a Dantzig-Wolfe reformulation of the MCM, while keeping either the demand constraints (4.2) (for the SBM) or the supply constraints (4.4) (for the CBM) in the master problem.




(a) The SBM (b) The CBM

Figure 4.2: The basic idea for each reformulation

### 4.2.2 The Supplier-Based Model

Let $T_i$ denote the set of all feasible flows from supplier $i$ to the $m$ customers with an accumulated flow less than or equal to $S_i$ and the flow on each arc $(i,j)$ less than or equal to $L_{ijq}$, i.e. $T_i = \{(x_{ij})_{j=1}^{m} : 0 \leq x_{ij} \leq L_{ijq}, \forall j \in J \text{ and } \sum_{j=1}^{m} x_{ij} \leq S_i\}$ . That is, each element of $T_i$ is a vector with $m$ entries and each entry $j$ represents the flow from $i$ to $j$. The flow on the arc

$(i,j)$ for element $t \in T_i$ is also denoted $x_{ij}^t$. For each flow in $T_i$, we define the associated costs of the flow by $C_i^t$. Now, the PLTP can be formulated as

$$
\begin{aligned}
\text{(SBM)} \quad & \min \sum_{i=1}^{n} \sum_{t \in T_i} \beta_i^t C_i^t, && && (4.9)\\
& \text{s.t.} \sum_{t \in T_i} \beta_i^t = 1, && \forall\, i \in I, && (4.10)\\
& \sum_{i=1}^{n} \sum_{t \in T_i} \beta_i^t x_{ij}^t = d_j, && \forall\, j \in J, && (4.11)\\
& \beta_i^t \in \{0,1\}, && \forall\, i \in I,\ \forall\, t \in T_i, && (4.12)
\end{aligned}
$$

The objective is to minimize the total cost as defined in equation (4.9). Constraints (4.10) state that for each supplier, exactly one flow in $T_i$ must be chosen, assuming that the all-zero flow belongs to $T_i$. The demand of each customer must be satisfied by constraints (4.11). Finally, all variables $\beta_i^t$ must be binary.

### 4.2.3 The Customer-Based Model

With a slight abuse of notation, define $T_j$ as the set of feasible flows from all suppliers to customer $j$ where the total flow is at least $d_j$, while respecting all arc capacities, i.e. $T_j = \{(x_{ij})_{i=1}^n : 0 \le x_{ij} \le L_{ijq}, \forall\, i \text{ and } \sum_{i=1}^n x_{ij} = d_j\}$. We still denote by $x_{ij}^t$ the flow from $i$ to $j$ for the element $t \in T_j$ and let $C_j^t$ denote the total cost implied by this element. Using these variables the PLTP can be stated as

$$
\begin{aligned}
\text{(CBM)} \quad & \min \sum_{j=1}^{m} \sum_{t \in T_j} \beta_j^t C_j^t, && && (4.13)\\
& \text{s.t.} \sum_{t \in T_j} \beta_j^t = 1, && \forall\, j \in J, && (4.14)\\
& \sum_{j=1}^{m} \sum_{t \in T_j} \beta_j^t x_{ij}^t \le S_i, && \forall\, i \in I, && (4.15)\\
& \beta_j^t \in \{0,1\}, && \forall\, j \in J,\ \forall\, t \in T_j. && (4.16)
\end{aligned}
$$

The objective is to minimize the total cost as defined in equation (4.13). For each customer one flow that satisfies his/her demand must be chosen, enforced by constraints (4.14). By equations (4.15) the capacity at the suppliers must be obeyed. All variables $\beta_i^t$ are binary.

Note, that for these formulations to be valid we make the assumption that there exists an integer-valued optimal solution in terms of the $(x, v)$-variables. This can be achieved by restricting all the $L_{ijl}$'s, $d_j$'s and $S_i$'s to integer numbers, which is an acceptable assumption given that rational data can always be transformed appropriately.

### 4.2.4 The Pricing Problems

The two formulations SBM and CBM contain a very large number of variables, namely a variable for each feasible integer flow from a supplier or to a customer. We use a column generation scheme to solve their LP relaxations denoted by SBM-LP and CBM-LP due to the large number of variables. In order for such a method to be efficient, it is crucial that the pricing problem can be efficiently solved. In the following we investigate the pricing problems for the SBM-LP and the CBM-LP. We find that the pricing problems faced have the same structure and can be solved by the same method.

#### Pricing Out Variables for the SBM-LP.

Let $\omega_i$ and $\eta_j$ denote the optimal dual variables associated with the constraints (4.10) and (4.11), respectively, in the current iteration of the restricted master problem. A variable $\beta_i^t$ prices out if

$$C_i^t - \omega_i - \sum_{j=1}^{m} \eta_j x_{ij}^t < 0, \tag{4.17}$$

The problem of identifying the variable with most negative reduced cost can thus be found by solving the following problem for each supplier $i$.

$$\text{(Pricing P}_i^{\text{SBM}}) \quad -\omega_i + \min \sum_{j=1}^{m} \sum_{l=1}^{q} ((c_{ijl} - \eta_j) x_{ijl} + g_{ijl} v_{ijl}), \tag{4.18}$$

$$\text{s.t.} \sum_{j=1}^{m} \sum_{l=1}^{q} x_{ijl} \leq S_i, \tag{4.19}$$

$$\sum_{l=1}^{q} v_{ijl} \leq 1, \qquad \forall j \in J, \tag{4.20}$$

$$x_{ijl} \leq L_{ijl} v_{ijl}, \qquad \forall (j,l), j \in J, l \in Q \tag{4.21}$$

$$x_{ijl} \geq L_{ij,l-1} v_{ijl}, \qquad \forall (j,l), j \in J, l \in Q \tag{4.22}$$

$$x_{ijl} \geq 0, \qquad \forall (j,l), j \in J, l \in Q \tag{4.23}$$

$$v_{ijl} \in \{0,1\}, \qquad \forall (j,l), j \in J, l \in Q \tag{4.24}$$

By adding a dummy customer this problem is readily recognized as a *Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem* (SSFCMCTP) for which an efficient algorithm based on dynamic programming is available from Christensen et al. (2012) and known to be $\mathcal{NP}$-Hard. Note, that only modes for which

$$(c_{ijl} - \eta_j) L_{ij,l-1} + g_{ijl} < 0, \tag{4.25}$$

will be considered in an optimal solution for the pricing problem, since a flow of all zeros is feasible.

**Pricing Out Variables for the CBM-LP.**

Let $\pi_j$ and $\lambda_i \leq 0$ denote the optimal dual variables associated now with the constraints (4.14) and (4.15). A variable $\beta_j^t$ prices out if

$$C_j^t - \pi_j - \sum_{i=1}^{n} \lambda_j x_{ij}^t < 0$$

For each customer $j$ the problem of finding the most negative reduced cost can then be formulated, as

$$
\begin{aligned}
&(\text{Pricing } \mathrm{P}_j^{\mathrm{CBM}}) \quad -\pi_j + \min \sum_{i=1}^{n} \sum_{l=1}^{q} ((c_{ijl} - \lambda_j) x_{ijl} + g_{ijl} v_{ijl}), && && (4.26)\\
&\text{s.t. } \sum_{i=1}^{n} \sum_{l=1}^{q} x_{ijl} = d_j, && && (4.27)\\
&\sum_{l=1}^{q} v_{ijl} \leq 1, && \forall i \in I, && (4.28)\\
&x_{ijl} \leq L_{ijl} v_{ijl}, && \forall (i,l), i \in I, l \in Q && (4.29)\\
&x_{ijl} \geq L_{ij,l-1} v_{ijl}, && \forall (i,l), i \in I, l \in Q && (4.30)\\
&x_{ijl} \geq 0, && \forall (i,l), i \in I, l \in Q && (4.31)\\
&v_{ijl} \in \{0,1\}, && \forall (i,l), i \in I, l \in Q. && (4.32)
\end{aligned}
$$

This problem is also a SSFCMCTP, which we solve by the algorithm of Christensen et al. (2012).

## 4.3 Strength of the LP Relaxation of the Models

In this section we investigate the theoretical strength of the new models compared to MCM which corresponds to a standard textbook way of representing piecwise linear cost functions. In addition we investigate the computational effectiveness of the models by presenting their actual strength on a subset of the instances used in Section 4.6.

### 4.3.1 Theoretical Strength

Not surprisingly, the LP relaxations of the CBM and SBM are at least as strong as that of the MCM, as these two models are Dantzig-Wolfe reformulations of the latter. Let $z(\cdot)$ denote the optimal solution value of the problem $(\cdot)$.

**Theorem 1**
*The following holds*

$$z(MCM\text{-}LP) \leq z(SBM\text{-}LP) \text{ and } z(MCM\text{-}LP) \leq z(CBM\text{-}LP), \tag{4.33}$$

*and there exist instances for which the inequalities are strict.*

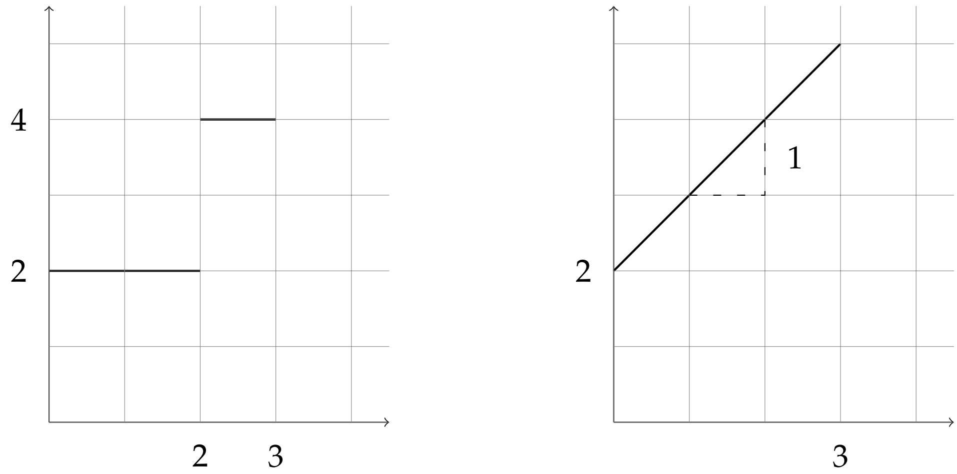
**Proof.** As the SBM is a Dantzig-Wolfe reformulation of the MCM, then z(SBM-LP) is equal to the optimal solution value found by a specific Lagrangean relaxation by Geoffrion (1974). It is a well-known fact that the bound found by applying any Lagrangean relaxation is at least as good as the bound found by the LP relaxation and the proof is completed (see e.g. Nemhauser and Wolsey (1999)). The same argument applies to the CBM. Below we present some instances that proves that the inequalities might be strict. ∎

This theoretical result does not directly prove that the new models will provide a strictly better bound than the LP relaxation on any instance. The computational results in the next section, however, show that this is very often the case on our test bed. Note, that there is no inclusion relation between the SBM and the CBM as the following examples show.

**An Example where z(SBM-LP) > z(CBM-LP) > z(MCM-LP).** Consider the following example with two suppliers and two customers. The suppliers capacities are $S_1 = 3$ and $S_2 = 4$ and the customer's demands are $d_1 = 3$ and $d_2 = 4$. The cost functions are specified in Figure 4.3 and assumed to be the same for either customer. Note, that $L_{22q} = L_{221} < \min\{S_2, d_2\} = 4$. The optimal solution to the z(SBM-LP) is

For supplier 1: Set the variable $\beta_1^1 = 1$ with the flow $\begin{pmatrix} 0 \\ 3 \end{pmatrix}$ and costs $C_1^1 = 4$.

For supplier 2: Set the variable $\beta_2^1 = 1$ with the flow $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ and costs $C_2^1 = 8$.




(a) Supplier 1 (b) Supplier 2

Figure 4.3: The cost function for suppliers 1 and 2, respectively, to either customer

Thus, z(SBM-LP)=12 which is the optimal (integer) solution. For the z(CBM-LP) the optimal solution is

$$\text{For customer 1: Set the variable } \beta_1^1 = 1/3 \text{ with flow } \begin{pmatrix} 3 \\ 0 \end{pmatrix} \text{ and costs } C_1^1 = 4,$$

$$\text{and the variable } \beta_1^2 = 2/3 \text{ with flow } \begin{pmatrix} 0 \\ 3 \end{pmatrix} \text{ and costs } C_1^2 = 5.$$

$$\text{For customer 2: Set the variable } \beta_2^1 = 1 \text{ with flow } \begin{pmatrix} 2 \\ 2 \end{pmatrix} \text{ and costs } C_2^1 = 6.$$

Hence, z(CBM-LP)=10+2/3, which is strictly less than z(SBM-LP). By comparison the LP relaxation of the MCM leads to the solution where

$$v_{111} = 1,\ v_{121} = 1/2,\ v_{211} = 1/3,\ v_{221} = 1 \tag{4.34}$$

$$x_{111} = 2,\ x_{121} = 1,\ x_{211} = 1,\ x_{221} = 3. \tag{4.35}$$

This solution has a cost of z(MCM-LP)=9+2/3. However, the CBM might provide a better LP relaxation than the SBM as illustrated in the following example.

**An Example where z(MCM-LP)=z(SBM-LP) < z(CBM-LP).** Consider the previous example, but modify the capacities and demands by setting $S_1 = S_2 = 4$, $d_1 = 2$, and $d_2 = 3$. One can verify that the optimal solution to the z(SBM-LP) is given by

$$\text{For supplier 1: Set the variable } \beta_1^1 = 1 \text{ with flow } \begin{pmatrix} 2 \\ 2 \end{pmatrix} \text{ and costs } C_1^1 = 4.$$

$$\text{For supplier 2: Set the variable } \beta_2^1 = 1/3 \text{ with flow } \begin{pmatrix} 0 \\ 3 \end{pmatrix} \text{ and costs } C_2^1 = 5,$$

$$\text{and the variable } \beta_2^2 = 2/3 \text{ with flow } \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ and costs } C_2^2 = 0,$$

and thereby the total costs are z(SBM-LP)=5+2/3. It can be verified that this solution is also the optimal solution to the z(MCM-LP). For CBM-LP we obtain the solution

$$\text{For customer 1: Set the variable } \beta_1^1 = 1 \text{ with flow } \begin{pmatrix} 2 \\ 0 \end{pmatrix} \text{ and costs } C_1^1 = 2.$$

$$\text{For customer 2: Set the variable } \beta_2^1 = 1/3 \text{ with flow } \begin{pmatrix} 0 \\ 3 \end{pmatrix} \text{ and costs } C_2^1 = 5,$$

$$\text{and variable } \beta_2^2 = 2/3 \text{ with flow } \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ and costs } C_2^2 = 0,$$

which means that z(CBM-LP)=6+1/3, which is strictly higher than z(SBM-LP).

### 4.3.2 Strength on Test Instances

To test the strength of each of the models, MCM, SBM and CBM, we measure the gap between the value of their LP relaxations and of the optimal solution of MCM. The gap is defined as $100 \cdot (OPT - z(\cdot))/OPT$ and the results are presented in Table 4.1. We confine these tests to the instances from Section 4.6 with 5 modes. Note that the time to solve the LP relaxations for the MCM and the CBM on all instances was less than 1 CPU second. For

Table 4.1: Average gap between the optimal solution and the LP relaxation value at the root node

| $n$ | $m$ | MCM | CBM | SBM |
|---|---|---|---|---|
| 5 | 15 | 7.85% | 0.51% | 7.37% |
| | 30 | 7.99% | 0.06% | 7.93% |
| | 50 | 7.28% | 0.03% | 7.25% |
| | 100 | 7.89% | 0.01% | 7.89% |
| 10 | 15 | 9.14% | 1.55% | 7.04% |
| | 30 | 7.90% | 0.33% | 7.49% |
| | 50 | 7.28% | 0.16% | 7.08% |
| | 100 | 7.07% | 0.04% | 7.04% |
| 15 | 15 | 9.24% | 2.45% | 6.92% |
| | 30 | 7.50% | 0.68% | 6.59% |
| | 50 | 6.89% | 0.27% | 6.53% |
| | 100 | 7.33% | 0.04% | 7.27% |

the SBM it took significantly longer to solve the LP relaxation and in some cases over 100 CPU seconds, as many more variables were generated compared to the CBM. These results indicate that the CBM outperforms the SBM by far and also the MCM with respect to closing the gap in the root node. As shown in the examples of Subsection 4.3.1 this relation does, however, not hold in general. In the instances considered in this paper the number of customers is always as least as high as the number of suppliers, i.e. $n \leq m$, since this is the case for most real-world distribution systems. Therefore, the number of constraints kept in the master problem is higher for the SBM than for the CBM, which might favor the CBM. Encouraged by these results we choose to develop our exact solution method by solving the CBM. The following two sections will explain how we add valid inequalities to the LP relaxation of CBM and the type of branching scheme we apply.

## 4.4 Valid Inequalities for the CBM

Though the solution to the current LP relaxation of the program (4.13)–(4.16) satisfies the capacity constraints, the $x_{ij}^t$ values of the columns used in this solution might not. To strengthen the formulation, we add *cover inequalities*, which are valid inequalities based on the knapsack structure of inequalities (4.15). For more information on cover inequalities see e.g. Nemhauser and Vance (1994); Gu et al. (1998, 1999); Wolsey (1998).

Let us define a vector $A = (a_j)_{j \in J}$ where $a_j \geq 0$, $\forall j$. We say that $A$ characterizes a *cover* if

$$\sum_{\{j: a_j \leq d_j\}} a_j > S_i, \tag{4.36}$$

for some $i \in I$. If $A$ characterizes a cover, then the so-called *cover inequality* is valid for the CBM

$$\sum_{\{(j,t): x_{ij}^t = a_j\}} \beta_j^t \leq C(A) - 1, \tag{4.37}$$

where $C(A) = |\{a_j : a_j \leq d_j\}|$. Further, the following strengthen inequality, called the *extended cover inequality*, is also valid for the CBM

$$\sum_{\{(j,t): x_{ij}^t \geq a_j\}} \beta_j^t \leq C(A) - 1. \tag{4.38}$$

**An Example where z(SBM-LP) > z(CBM-LP) > z(MCM-LP), Revisited.** Consider, again, the following optimal LP solution to the CBM

$$\text{For customer 1: Set the variable } \beta_1^1 = 1/3 \text{ with flow } \begin{pmatrix} 3 \\ 0 \end{pmatrix} \text{ and costs } C_1^1 = 4,$$

$$\text{and the variable } \beta_1^2 = 2/3 \text{ with flow } \begin{pmatrix} 0 \\ 3 \end{pmatrix} \text{ and costs } C_1^2 = 5.$$

$$\text{For customer 2: Set the variable } \beta_2^1 = 1 \text{ with flow } \begin{pmatrix} 2 \\ 2 \end{pmatrix} \text{ and costs } C_2^1 = 6.$$

The vector $\begin{pmatrix} 3 & 2 \end{pmatrix}^T$ characterizes a cover since

$$3 + 2 = a_1 + a_2 > S_2 = 4. \tag{4.39}$$

The corresponding extended cover inequality implied is

$$\sum_{\{t \in T_1 : x^t_{21} \geq 3\}} \beta^t_1 + \sum_{\{t \in T_2 : x^t_{22} \geq 2\}} \beta^t_2 \leq 1. \tag{4.40}$$

Note, that the addition of this valid inequality will cut off the optimal LP solution.

### 4.4.1 How to Separete Cover Inequalities

Given a solution $(\tilde{\beta}^t_j)$ to the current LP, an extended cover inequality characterized by some vector $A$ is violated if $\sum_{j=1}^m (1 - \sum_{t : x^t_{ij} \geq a_j} \beta^t_j) \leq 1$. Then, consider variables $y(a_j)$ taking value one if the $j$-th coordinate of vector A is equal to $a_j$ and zero otherwise. Finding a vector $A$ characterizing an inequality which is most violated by a current solution $(\tilde{\beta}^t_j)$ can thus be formulated as follows.

$$\min \sum_{j=1}^{m} \sum_{a_j=0}^{d_j} \Big(1 - \sum_{t : x^t_{ij} \geq a_j} \tilde{\beta}^t_j\Big) y(a_j), \tag{4.41}$$

$$\text{s.t.} \ \sum_{j=1}^{m} \sum_{a_j=0}^{d_j} a_j y(a_j) \geq S_i + 1, \tag{4.42}$$

$$\sum_{a_j=0}^{d_j} y(a_j) \leq 1, \qquad \forall j \in J, \tag{4.43}$$

$$y(a_j) \in \{0, 1\}, \qquad \forall j \in J,\ \forall a_j = 0, \ldots, d_j. \tag{4.44}$$

Further, notice that it is only necessary to consider values $a_j$ such that, in the current solution, there is a positive column whose flow $x^t_{ij} = a_j$. In other words, if we define $\tilde{A}_j = \{x^t_{ij} : \tilde{\beta}^t_j > 0\}$, the separation problem is

$$
\begin{aligned}
(\text{EC}_i \text{ Sep.}) \quad & \min \sum_{j=1}^{m} \sum_{a_j \in \tilde{A}_j} \Big(1 - \sum_{t: x_{ij}^t \geq a_j} \tilde{\beta}_j^t\Big) y(a_j), && && (4.45)\\
& \text{s.t.} \sum_{j=1}^{m} \sum_{a_j \in \tilde{A}_j} a_j y(a_j) \geq S_i + 1, && && (4.46)\\
& \sum_{a_j \in \tilde{A}_j} y(a_j) \leq 1, && \forall j \in J, && (4.47)\\
& y(a_j) \in \{0, 1\}, && \forall j \in J,\ \forall a_j \in \tilde{A}_j. && (4.48)
\end{aligned}
$$

This program ($\text{EC}_i$ Sep.) is a *Multiple-Choice Knapsack Problem* (MCKP) for which a specialized algorithm is available, see Pisinger (1995).

We initialize the current LP with CBM-LP. Then, given a solution $(\tilde{\beta}_j^t)$ to the current LP, we identify violated *extended cover inequalities*, add them to this current LP and solve it again. The added inequalities can be handled very efficiently in our subproblem as explained in the following subsection.

### 4.4.2 How to Deal with Cover Inequalities in the Pricing Problem

When using column generation the addition of valid inequalities is a delicate task. It is crucial that the added inequalities do not destroy the structure of the pricing problem. Fortunately, a cover inequality is easy to incorporate into the pricing problem for the CBM. Let $\mathcal{A}_i$ denote the set of vectors characterizing the extended cover inequalities that were derived from the capacity constraint (4.15) for supplier $i$ and currently added to the (restricted) master problem, i.e.

$$
\sum_{\{(j,t): x_{ij}^t \geq a_j\}} \beta_j^t \leq C(A) - 1, \qquad \forall A \in \mathcal{A}_i, \forall i \in I. \tag{4.49}
$$

Thus, the restricted master problem to be solved is $\min\{(4.13) \text{ s.t. } (4.14), (4.15), (4.49), \beta_j^t \geq 0, \forall j \in J, \forall t \in T_j\}$.

If we denote by $\zeta_A$ the non-positive dual variable associated to the extended cover inequality characterized by $A$, a variable $\beta_j^t$ now prices out if

$$C_j^t - \pi_j - \sum_{i=1}^{n} \lambda_j x_{ij}^t - \sum_{i=1}^{n} \sum_{A \in \mathcal{A}_i : x_{ij}^t \geq a_j} \zeta_A < 0.$$

Hence, for each customer $j$, the objective function (4.26) of problem (Pricing $\mathrm{P}_j^{\mathrm{CBM}}$) must be replaced by

$$-\pi_j + \min \sum_{i=1}^{n} \left( \sum_{l=1}^{q} ((c_{ijl} - \lambda_j) x_{ijl} + g_{ijl} v_{ijl}) - \sum_{A \in \mathcal{A}_i : x_{ij}^t \geq a_j} \zeta_A \right). \tag{4.50}$$

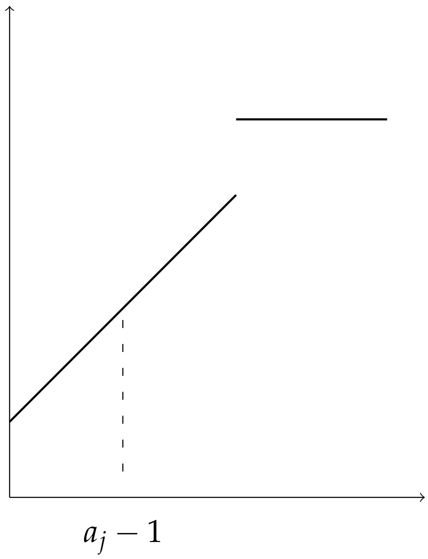
This modified pricing problem can still be seen as a Single-Sink, Fixed-Charge, Multiple-Choice Transportation Problem and can thus be solved using the algorithm of Christensen et al. (2012) if additional breakpoints (or modes) are inserted and fixed and variable costs are adapted as follows.

First, for each arc $(i, j)$, we add a breakpoint to the cost function for each different value $a_j - 1$. Let $Q_{ij}^{new}$ represent the set of indices of the added breakpoints for arc $(i, j)$. We denote any breakpoint by $L_{ijl}$, but if $l \in Q$, it is an original breakpoint and if $l \in Q_{ij}^{new}$, then it is a new breakpoint, in which case it is equal to some value $a_j - 1$.

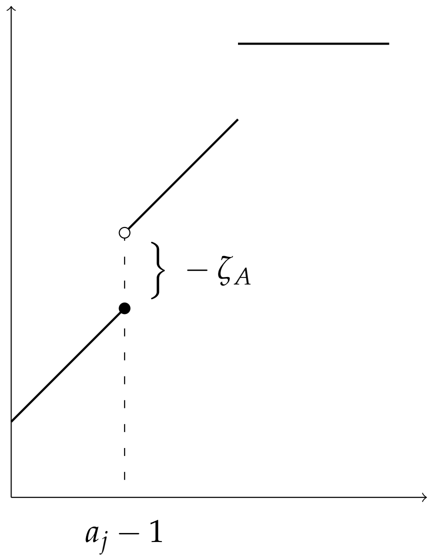
Then, if $l \in Q$, its variable cost $c_{ijl}$ is unchanged and its fixed cost is replaced by $g_{ijl} - \sum_{A : a_j - 1 \leq L_{ijl}} \zeta_A$.

Finally, if $l \in Q_{ij}^{new}$, let $l' \in Q$ be the index of the breakpoint $L_{ijl'}$ such that $L_{ij,l'-1} < L_{ijl} \leq L_{ij,l'}$. The variable cost corresponding to this breakpoint $L_{ijl}$ is set equal to $g_{ij,l'-1}$ and its variable cost to $g_{ij,l'-1} - \sum_{A : a_j \leq L_{ijl}} \zeta_A$.

The interpretation of this addition of breakpoints and cost adaptation is the following. When pricing out a new variable which has a flow greater than or equal to some $a_j$, then the variable appears in the corresponding equation (4.49) and the dual variable $\zeta_j$ should be subtracted when calculating its reduced cost. This is implemented by stating that for each element $A$ in $\mathcal{A}_j$, we introduce a new mode at $a_j - 1$ which has the same variable cost and upper capacity limit as the original mode, but the fixed cost of this and all subsequent modes will be decreased by $\zeta_A$ , i.e. increased by $-\zeta_A$ which is non-negative (see Figure 4.4). Hence, the subproblem solved will very often contain more modes than the original problem due to added cover inequalities.




(a) Original Cost Function




(b) Modified Cost Function

Figure 4.4: How to incorporate a cover inequality with the dual variable $\zeta_j$ and entry $a_j$ in the characterizing vector for customer $j$.

That is incorporating the valid inequalities defined by constraint (4.49) from the master problem into the subproblem reduces to manipulating the objective function of the pricing problem by adding a number of new variables.

## 4.5 Branching Rule

Any solution, $\tilde{\beta}_j$, to the CBM corresponds to a solution for the MCM, given by

$$\tilde{v}_{ijl} = \sum_{t:v^t_{ijl}=1} \tilde{\beta}^t_j \quad \text{and} \quad \tilde{x}_{ijl} = \sum_{t:v^t_{ijl}=1} x^t_{ijl}\tilde{\beta}^t_j. \tag{4.51}$$

A solution is integer if and only if no $\tilde{v}_{ijl}$ is fractional. This insight is crucial in developing branching rules for the CBM. In preliminary tests a number of different approaches were tested and based on these results we employ the following branching rule. Given the current LP solution, calculate the cost on every arc as

$$\tilde{C}_{ij} = \sum_{l=1}^{q} \tilde{x}_{ijl} c_{ijl} + \tilde{v}_{ijl} g_{ijl}. \tag{4.52}$$

Then calculate the difference between $\tilde{C}_{ij}$ and the real cost of a flow of $\sum_{l=1}^{q} \tilde{x}_{ijl}$ units on the arc. Branching is done by selecting the arc with the largest difference and the mode that corresponds to this flow. When a variable to branch on, say $\tilde{v}_{\bar{i}\bar{j}\bar{l}}$, has been chosen, the branching is done as follows. On the to-zero branch we require that

$$\sum_{l=\bar{l}}^{q} v_{\bar{i}\bar{j}l} = 0 \tag{4.53}$$

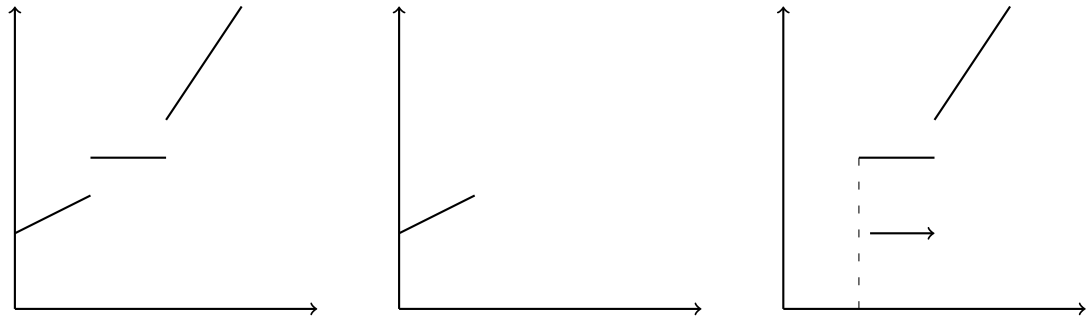
and on the to-one branch that

$$\sum_{l=\bar{l}}^{q} v_{\bar{i}\bar{j}l} = 1. \tag{4.54}$$

This has a nice result of either limiting the flow on an arc to at most $L_{ij,l-1}$ (on the to-zero branch) or forcing it to be at least $L_{ij,l-1}$ (on the to-one branch), which is depicted in Figure 4.5. This kind of restriction is easy to handle in the pricing problem. This is known as branching on a *GUB Dichotomy*, a term from Linderoth and Savelsbergh (1999). To select the next node to be processed a *hybrid estimate* selection scheme was used (see Achterberg (2009)).

## 4.6 Computational Tests

In this section we describe how a number of test instances were generated and how the proposed method compares to the standard solver CPLEX (version 12.4).



(a) $\overline{v}_{ij2}$ is fractional (parent node) (b) The to-zero node (c) The to-one node

Figure 4.5: An illustration of the branching scheme.

### 4.6.1 Generation of Test Instances

We generate a number of test instances in the following way. First of all we consider instances having $n \in \{5, 10, 15\}$ suppliers and $m \in \{15, 30, 50, 100\}$ customers, while the number of modes is $\{3, 5, 7\}$. Let $U[x, y]$ denote the uniform distribution from $x$ to $y$ and $UI$ the discretized version. The customer demand is then drawn from $UI[20, 100]$ with expected value 60. The supplier capacity is drawn from $UI[0.8 \cdot k, 1.2 \cdot k]$, where $k = 60m/n$, hence $k$ is the average demand per supplier. The increase in capacity on an arc is distributed as $UI[0.8 \cdot l, 1.2 \cdot l]$ for each mode, while the fixed cost increase is drawn from $UI[2, 10]$. The variable cost for mode $(i, j, l)$ is calculated as $U[0, (g_{ijl} - g_{ij,l-1})/(L_{ijl} - L_{ij,l-1})]$ and rounded to two decimals. In this way there is a relationship between the fixed cost for using a mode and the capacity of the mode. These instances will have non-decreasing costs as well as integral capacities and demands. Furthermore, the total capacity on an arc is evenly spread among the modes.

In Vielma et al. (2010) a number of instances with 10 suppliers, 10 customers and different numbers of modes were presented, and they were also used in Keha et al. (2006) and Vielma et al. (2008). We also test the proposed solution method on the 100 instances with 4 modes as these are the only instances with integral mode capacities from that test bed. The remaining instances have highly fractional mode capacities, which makes a transformation into instances with integral mode capacities nontrivial. Furthermore, these instances might have decreasing costs, which means that the pricing algorithm of Christensen et al. (2012) , which assumes a non-decreasing cost structure, cannot be applied as a subroutine. On these instances, we use a straightforward, naive dynamic programming algorithm to solve the pricing problems instead.

### 4.6.2 Computational Results

In order to test the proposed method, we compare it on a number of test instances to the MCM passed to the general solver CPLEX (version 12.4). The tests were conducted on a laptop with a 2.4 GHz Intel Core 2 Duo CPU and 4 GB of RAM, running Linux and using the `gcc` compiler. The proposed method was implemented using the branch-cut-and-price framework SCIP (version 2.1.1, see Achterberg (2009) and SCIP (2010)). We used the concert technology interface for CPLEX and all parameters were set to default except that only one core was used, since SCIP does not currently support multiple cores. The different choices of suppliers, customers, and modes are detailed above and for each configuration 5 instances were generated. For each configuration we present the average time used to solve the five instances, the average number of nodes in the branching tree, and the optimality gap in the root node. The latter is presented both with and without the cuts presented in Section 4.4 for the CBM and for CPLEX with and without the different default cuts. We enforce a time limit of 3 hours of CPU time (10.800 secs). A number in parenthesis after the runtime indicates the number of instances that could not be solved to optimality within the time limit and a number in brackets indicates the number of instances for which the algorithm ran out of memory. In either case the instance was not included in the statistics. The number in bold indicates the fastest average run time.

In general the small instances of Table 4.2 can be solved rather fast by both methods, with the CBM being the fastest on all instances.

In Table 4.3 we see that CPLEX is consistently slightly faster than the CBM when there

Table 4.2: Computational results for instances with 5 suppliers.

| | | CPLEX | | | | CBM | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | w/o. cuts | w. cuts | | | w/o. cuts | w. cuts |
| $m$ | $q$ | secs. | #nodes | gap (%) | gap (%) | secs. | #nodes | gap (%) | gap (%) |
| 15 | 3 | 0.41 | 131.0 | 11.52 | 0.47 | **0.20** | 19.4 | 0.52 | 0.18 |
| | 5 | 1.67 | 672.0 | 7.85 | 0.69 | **0.63** | 42.6 | 0.51 | 0.24 |
| | 7 | 4.99 | 1667.6 | 6.79 | 0.60 | **0.74** | 45 | 0.36 | 0.18 |
| 30 | 3 | 0.60 | 120.4 | 11.94 | 0.24 | **0.59** | 36.6 | 0.09 | 0.06 |
| | 5 | 2.47 | 738.0 | 7.99 | 0.28 | **0.38** | 28.2 | 0.06 | 0.03 |
| | 7 | 4.61 | 1011.2 | 5.92 | 0.20 | **0.60** | 50.4 | 0.05 | 0.03 |
| 50 | 3 | 1.24 | 206.0 | 10.83 | 0.12 | **0.65** | 34.8 | 0.04 | 0.03 |
| | 5 | 7.17 | 1129.8 | 7.28 | 0.21 | **0.68** | 27.2 | 0.03 | 0.02 |
| | 7 | 20.13 | 2024.8 | 6.46 | 0.32 | **1.00** | 38.2 | 0.02 | 0.02 |
| 100 | 3 | 5.05 | 567.2 | 10.71 | 0.13 | **2.98** | 130.8 | 0.01 | 0.01 |
| | 5 | 13.48 | 1290.80 | 7.89 | 0.14 | **1.47** | 54.0 | 0.01 | 0.00 |
| | 7 | 60.87 | 3970.6 | 5.93 | 0.26 | **3.14** | 83.0 | 0.01 | 0.01 |

Table 4.3: Computational results for instances with 10 suppliers.

| | | CPLEX | | | | CBM | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | w/o. cuts | w. cuts | | | w/o. cuts | w. cuts |
| $m$ | $q$ | secs. | #nodes | gap (%) | gap (%) | secs. | #nodes | gap (%) | gap (%) |
| Vielma et al. (2010) | | | | | | | | | |
| 10 | 4 | 1.63 | 610.69 | 4.52 | 1.30 | **0.52** | 10.52 | 2.52 | 0.23 |
| 15 | 3 | **5.59** | 1654.8 | 12.56 | 1.22 | 18.94 | 236.4 | 1.72 | 0.62 |
| | 5 | 25.34 | 5535.8 | 9.14 | 1.35 | **23.37** | 302.8 | 1.55 | 0.60 |
| | 7 | 28.22 | 4861.2 | 6.38 | 1.03 | **2.63** | 92.2 | 0.59 | 0.20 |
| 30 | 3 | **4.74** | 935.6 | 10.57 | 0.52 | 10.76 | 401.2 | 0.42 | 0.18 |
| | 5 | 24.62 | 3499.8 | 7.90 | 0.54 | **5.68** | 132.2 | 0.33 | 0.17 |
| | 7 | 207.85 | 28,548.4 | 5.25 | 0.53 | **15.40** | 702.0 | 0.23 | 0.16 |
| 50 | 3 | **9.67** | 2001.2 | 10.83 | 0.28 | 12.07 | 572.6 | 0.15 | 0.10 |
| | 5 | 190.62 | 23,554.4 | 7.28 | 0.49 | **62.08** | 2,603.4 | 0.16 | 0.12 |
| | 7 | 341.92 | 31,550.6 | 5.50 | 0.48 | **13.52** | 471.6 | 0.11 | 0.08 |
| 100 | 3 | **28.23** | 3,016.6 | 11.89 | 1.43 | 38.8 | 1,000.0 | 0.05 | 0.05 |
| | 5 | 263.94 | 17,485.4 | 7.07 | 0.32 | **76.01** | 1,789.4 | 0.04 | 0.03 |
| | 7 | 262.22 | 14,723.2 | 5.23 | 0.26 | **24.93** | 656.6 | 0.01 | 0.01 |

are only 3 modes. The CBM exhibits a slightly unexpected behavior when looking at the run-times from 5 to 7 modes. On these instances the run times often decrease, which is counterintuitive. This might be explained by the simultaneous decrease in the optimality gap. The instances from Vielma et al. (2010) seem to be comparable in terms of the run-time for both solution methods. The expected decrease in performance for the CBM due to the naive dynamic programming algorithm being used for pricing is hardly noticeable, probably because of the relatively small instance size.

Table 4.4: Computational results for instances with 15 suppliers.

| | | CPLEX | | | | CBM | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | w/o. cuts | w. cuts | | | w/o. cuts | w. cuts |
| $m$ | $q$ | secs. | #nodes | gap (%) | gap (%) | secs. | #nodes | gap (%) | gap (%) |
| 15 | 3 | **8.88** | 2,315.4 | 12.98 | 1.77 | 12.66 | 262.8 | 3.38 | 0.67 |
| | 5 | 136.99 | 18,105.4 | 9.24 | 1.69 | **46.93** | 495.6 | 2.45 | 0.58 |
| | 7 | 413.95 | 33,696.8 | 6.40 | 1.16 | **58.17** | 362.8 | 1.36 | 0.44 |
| 30 | 3 | **21.14** | 4,098.8 | 11.81 | 0.72 | 35.39 | 698.4 | 0.73 | 0.21 |
| | 5 | 546.29 | 41,569.8 | 7.50 | 0.81 | **287.48** | 3,736.2 | 0.68 | 0.32 |
| | 7 | 3231.53 | 198,418.8 | 5.54 | 0.84 | **201.83** | 2,885.0 | 0.50 | 0.25 |
| 50 | 3 | **55.32** | 5,707.4 | 10.97 | 1.34 | 56.74 | 1,306.6 | 0.34 | 0.15 |
| | 5 | 785.28 | 56,559.2 | 6.89 | 0.52 | **149.83** | 3,326.8 | 0.27 | 0.16 |
| | 7 | 3,441.36 (2) | 269,393.67 | 5.11 | 0.61 | **586.62** | 4,790.6 | 0.15 | 0.11 |
| 100 | 3 | **57.87** | 5,822.8 | 10.04 | 0.47 | 60.94 | 1,294.0 | 0.29 | 0.27 |
| | 5 | 937.21 | 106,107.8 | 7.33 | 0.35 | **173.77** | 3,191.0 | 0.04 | 0.03 |
| | 7 | 7,203.71 (3) [1] | 279,870.0 | 4.88 | 0.38 | **594.34** | 9,270.6 | 0.05 | 0.03 |

In Table 4.4 we again see that the CBM is slower than CPLEX when there are only 3 modes. On the largest set of instances CPLEX fails to solve all instances, except one. The CBM solves all the instances and much faster.

In the Figures 4.6, 4.7, and 4.8 we plot the logarithm to the total solution time for each configuration of both the CBM and CPLEX for the values $q = 3$, $q = 5$, and $q = 7$, respectively, to facilitate comparison. For $q = 3$ there is little difference between using CPLEX and the CBM, but the CBM becomes highly favorable as $q$ increases. We only include the instances that could be solved within the imposed time limit and do not include the unsolved instances, which explains why the total solution time for CPLEX seems to be decreasing on the two largest instances in Figure 4.8. However, for the largest set of instances, CPLEX can only solve one of the five instances and even uses more time than the CBM does for solving all instances. As regards the second largest set of instances CPLEX fails to solve two out of the five instances. If we were to penalize the unsolved instances in Figure 4.8, the observed downwards trend for the biggest instances would be changed to a dramatic increase instead.

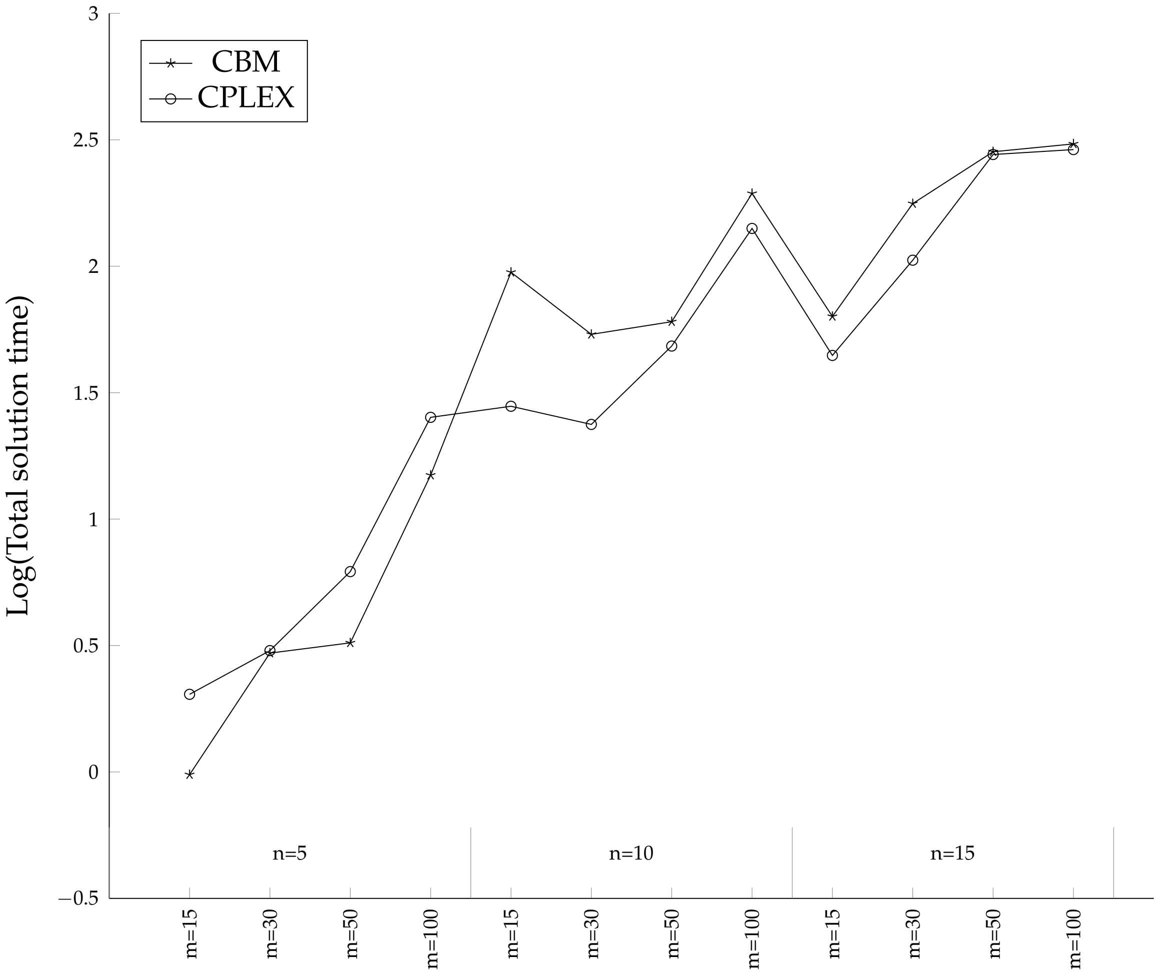



Figure 4.6: The logarithm to the total solution time for the CBM and CPLEX plotted against the different configurations where $q = 3$.

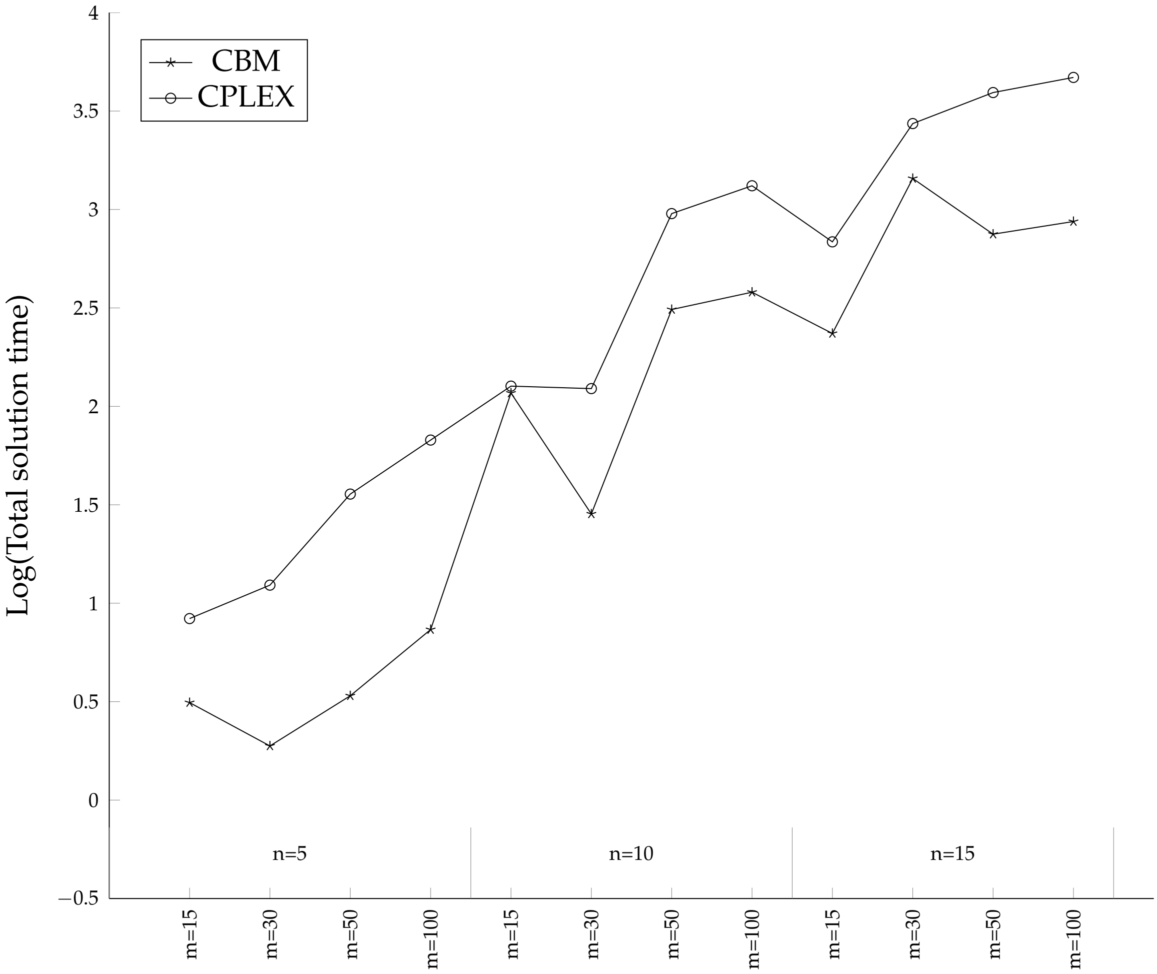



Figure 4.7: The logarithm to the total solution time for the CBM and CPLEX plotted against the different configurations where $q = 5$.

When we inspect the optimality gap for the reported instances, we find a noticeable pattern that warrants a few comments. To display this pattern we plot the gap between the optimal LP value and the optimal integer solution for the MCM and the CBM, without adding any cuts, against the ratio $m/n$. We take a simple average over the gaps in the two cases where the ratio of $m/n$ could refer to two different configurations as e.g. the instances with 5 suppliers and 15 customers and those with 10 suppliers and 30 customers, both have a ratio of $m/n = 3$. We restrict our comparison to the instances where $q = 3$, but the

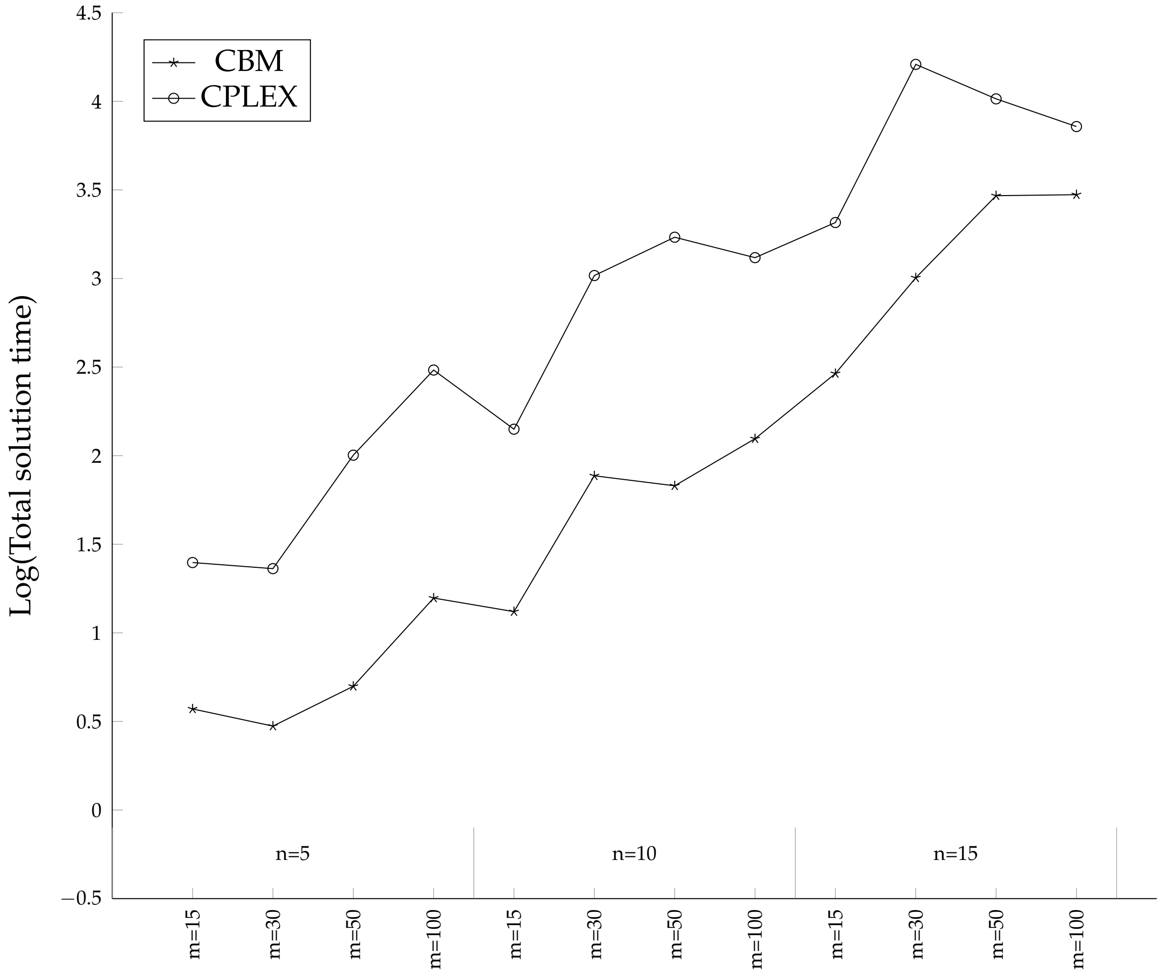



Figure 4.8: The logarithm to the total solution time for the CBM and CPLEX plotted against the different configurations where $q = 7$.

results are similar for the other settings of $q$. There seems to be a trend of the gap decreasing when the $m/n$-ratio increases. For the MCM this decline is roughly linear while it seems to be roughly exponential for the CBM. Here we summarize a few observations from these computational tests:

- When there are more than 3 modes, the CBM is always faster and on large instances it is significantly faster.
- The CBM results in much fewer nodes and a smaller optimality gap than the MCM.

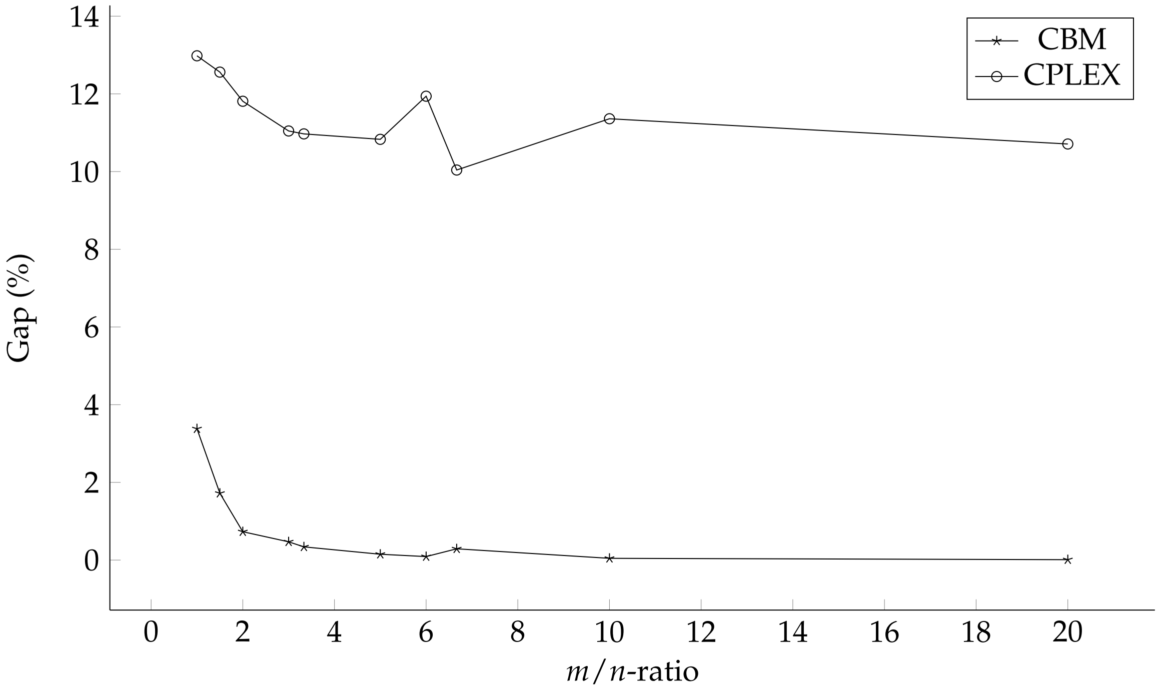



Figure 4.9: The optimality gap for the MCM and the CBM w/o cuts vs. the $m/n$-ratio for $q = 3$.

- While CPLEX is very effective in adding cuts, the gap of the CBM is always tighter after addition of the valid inequalities.
- There seems to be a relationship between the optimality gap and the ratio $m/n$.

## 4.7 Conclusion

In this paper we proposed two new formulations for the piecewise linear transportation problem. Both formulations have a possibly stronger LP relaxation bound than the standard models considered so far in the literature. In particular, the formulation termed the customer-based model seems to offer a significant reduction in the gap between the LP relaxation of the root node and the optimal solution. We extended this model into an exact solution method by adding valid inequalities and branching on the original variables. Tests indicate that this new solution method offers a sustained significant improvement over the standard solver of CPLEX on a test bed of randomly generated data with a reduction in run-time of 91.7% on the largest instances. Future research might focus on the addition of more valid inequalities and new branching rules. Also, further enhancements to the special case of one mode, the fixed-charge transportation problem, should be of interest.

# Bibliography


Achterberg, T. (2009). Scip: solving constraint integer programs. *Mathematical Programming Computation 1*(1), 1–41.

Alidaee, B. and G. A. Kochenberger (2005). A note on a simple dynamic programming approach to the single-sink fixed-charge transportation problem. *Transportation Science 39*(1), 140–143.

Amiri, A. (1997). Solution procedures for the service system design problem. *Computers & Operations Research 24*(1), 49–60.

Andonov, R., V. Poirriez, and S. Rajopadhye (2000). Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research 123*(2), 394–407.

Averbakh, I., O. Berman, Z. Drezner, and G. O. Wesolowsky (1998). The plant location problem with demand-dependt setup costs and centralized allocation. *European Journal of Operational Research 111*(3), 543–554.

Averbakh, I., O. Berman, Z. Drezner, and G. O. Wesolowsky (2007). The uncapacitated facility location problem with demand-dependent setup and service costs and customer-choice allocation. *European Journal of Operational Research 179*(3), 956–967.

Baumgartner, K., A. Fuetterer, and U. W. Thonemann (2012). Supply chain design considering economies of scale and transport frequencies. *European Journal of Operational Research 218*(3), 789–800.

Beasley, J. E. (1993). Lagrangean heuristics for location problems. *European Journal of Operational Research 65*(3), 383–399.

Bilde, O. and J. Krarup (1977). Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics 1*(1), 79–97.

Borgen, E., K. O. Schea, F. Rømo, and A. Tomasgard (2000). Optimal lokalisering av slakterianlegg. Technical Report STF38 F00609, SINTEF Industrial Management.

Broek, J. v., P. Schütz, L. Stougie, and A. Tomasgard (2006). Location of slaughterhouses under economies of scale. *European Journal of Operational Research 175*(2), 740–750.

Cañavate-Bernal, R., M. Landete-Ruiz, and A. Marín-Pérez (2000). On the resolution of the single product capacitated machine siting problem. *Journal of the Operational Research Society 51*(8), 982–992.

Christensen, T. R. L., K. A. Andersen, and A. Klose (2012). Solving the single-sink, fixed-charge, multiple-choice transportation problem by dynamic programming. To appear in *Transportation Science*, DOI: 10.1287/trsc.1120.0431.

Cornuejols, A., R. Sridharan, and J. M. Thizy (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research 50*(3), 280–297.

Correia, I. and M. E. Captivo (2003). A Lagrangean heuristic for a modular capacitated location problem. *Annals of Operations Research 122*(10), 141–161.

Correia, I. and M. E. Captivo (2006). Bounds for the single source modular capacitated plant location problem. *Computers & Operations Research 33*(1–4), 2991–3003.

Correia, I., L. Gouveia, and F. Saldanha-da Gama (2010). Discretized formulations for capacitated location problems with modular distribution costs. *European Journal of Operational Research 204*(2), 237–244.

Croxton, K. L., B. Gendron, and T. L. Magnanti (2003a). A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science 49*(9), 1268–1273.

Croxton, K. L., B. Gendron, and T. L. Magnanti (2003b). Models and methods for merge-in-transit operations. *Transportation Science 37*(1), 1–22.

Croxton, K. L., B. Gendron, and T. L. Magnanti (2007). Variable disaggregation in network flow problems with piecewise linear costs. *Operations Research 55*(1), 146–157.

Dasci, A. and V. Verter (2001). The plant location and technology acquisition problem. *IIE Transactions 33*(11), 963–973.

Daskin, M. S., C. R. Coullard, and Z.-J. M. Shen (2002). An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research 110*(1–4), 83–106.

Davenport, A. J. and J. Kalagnanam (2001, May). Price negotiations for procurement of direct inputs. Technical report, IBM T. J. Watson Research Center.

Duhamel, C., L. Gouveia, P. Moura, and M. de Souza (2012). Models and heuristics for the k -degree constrained minimum spanning tree problem with node-degree costs. *Networks 60*(1), 1–18.

Dupont, L. (2008). Branch and bound algorithm for a facility location problem with concave site dependent costs. *International journal of production economics 112*(1), 245–254.

Efroymson, M. A. and T. L. Ray (1966). A branch-bound algorithm for plant location. *Operations Research 14*(3), 361–386.

Elhedhli, S. (2006). Service system design with immobile servers, stochastic demand, and congestion. *Manufacturing & Service Operations Management 8*(1), 92–97.

Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research 26*(6), 992–1009.

Feldman, E., F. A. Lehrer, and T. L. Ray (1966). Warehouse location under continuous economies of scale. *Management Science 12*(9), 670.

Gabor, A. F. and J.-K. C. W. van Ommeren (2006). Approximation algorithms for facility location problems with a special class of subadditive cost functions. *Theoretical Computer Science 363*(3), 289–300.

Geoffrion, A. M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Studies 2*(2), 82–114.

Gu, Z., G. L. Nemhauser, and M. W. P. Savelsbergh (1998). Lifted cover inequalities for 0–1 integer programs: Computation. *INFORMS Journal on Computing 10*(4), 427 – 437.

Gu, Z., G. L. Nemhauser, and M. W. P. Savelsbergh (1999). Lifted cover inequalities for 0–1 integer programs: Complexity. *INFORMS Journal on Computing 11*(1), 117–123.

Guignard, M. and S. Kim (1987). Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical Programming 39*(2), 215–228.

Haberl, J. (1991). Exact algorithm for solving a special fixed-charge linear programming problem. *Journal of Optimization Theory and Applications 69*(3), 489–529.

Hajiaghayi, M. T., M. Mahdian, and V. Mirrokni (2003). The facility location problem with general cost functions. *Networks 42*(1), 42–47.

Hamacher, H. W. and S. Nickel (1998). Classification of location models. *Location Science 6*(1–4), 229–242.

Harkness, J. and C. ReVelle (2002). A Lagrangean heuristic for the staircase cost facility location problem. *International Journal of Industrial Engineering 9*(1), 36–44.

Harkness, J. and C. ReVelle (2003). Facility location with increasing production costs. *European Journal of Operational Research 145*(1), 1–13.

Held, M., P. Wolfe, and H. P. Crowder (1974). Validation of subgradient optimization. *Mathematical Programming 6*(1), 62–88.

Henderson, J. M. and R. E. Quandt (1971). *Mircroeconomic Theory: A Mathematical Approach*. New York, NY, USA: McGraw-Hill.

Herer, Y. T., M. J. Rosenblatt, and I. Hefter (1996). Fast algorithms for single-sink fixed charge transportation problems with applications to manufacturing and transportation. *Transportation Science 30*(4), 276–290.

Holmberg, K. (1989). Transportation and location problems with staircase costs. Technical report, Linköping Institute of Technology.

Holmberg, K. (1994). Solving the staircase cost facility location problem with decomposition and piecewise linearization. *European Journal of Operational Research 75*(1), 41–61.

Holmberg, K. (1999). Exact solution methods for uncapacitated location problems with convex transportation costs. *European Journal of Operational Research 114*(1), 127–140.

Holmberg, K. and J. Ling (1997). A Lagrangean heuristic for the facility location problem with staircase costs. *European Journal of Operational Research 97*(1), 63–74.

Holmberg, K. and H. Tuy (1999). A production-transportation problem with stochastic demand and concave production costs. *Mathematical Programming 85*(1), 157–179.

Kameshwaran, S. and Y. Narahari (2009). Nonconvex piecewise linear knapsack problems. *European Journal of Operational Research 192*(1), 56–68.

Keha, A. B., I. de Farias Jr., and G. L. Nemhauser (2004). Model for representing piecewise linear cost functions. *Operations Research Letters 32*(1), 44–48.

Keha, A. B., I. de Farias Jr., and G. L. Nemhauser (2006). A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research 54*(5), 847–858.

Kellerer, H., U. Pferschy, and D. Pisinger (2003). *Knapsack Problems*. Berlin, Germany: Springer.

Kelly, D. L. and B. M. Khumawala (1982). Capacitated warehouse location with concave costs. *The Journal of the Operational Research Society 33*(9), 817–826.

Kim, D. and P. M. Pardalos (1999). A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters 24*(4), 195–203.

Kim, D. and P. M. Pardalos (2000). A dynamic domain contraction algorithm for nonconvex piecewise linear network flow problems. *Journal of Global Optimization 17*(1–4), 225–234.

Klose, A. (2008). Algorithms for solving the single-sink fixed-charge transportation problem. *Computers & Operations Research 35*(6), 2079–2092.

Klose, A. and A. Drexl (2005a). Facility location models for distribution system design. *European Journal of Operational Research 162*(1), 4–29.

Klose, A. and A. Drexl (2005b). Lower bounds for the capacitated facility location probl. *Management Science 51*(11), 1689–1705.

Krarup, J. and P. M. Pruzan (1983). The simple plant location problem: Survey and synthesis. *European Journal of Operational Research 12*(1), 36–81.

Kubo, M. and H. Kasugai (1991). A lagrangean approach to the facility location problem with concave costs. *Journal of Operations Research Society of Japan 34*(2), 125–136.

Kuehn, A. A. and M. J. Hamburger (1963). A heuristic program for locating warehouses. *Management Science 9*(4), 643–666.

Lamar, B. W. (1993). A method for solving network flow problems with general nonlinear arc costs. In D.-Z. Du and P. M. Pardalos (Eds.), *Network Optimization Problems*, pp. 147–167. Singapore, Singapore: World Scientific Publishing Co.

Lapierre, S., A. B. Ruiz, and P. Soriano (2004). Designing distribution networks: Formulations and solution heuristic. *Transportation Science 38*(2), 174–187.

Linderoth, J. T. and M. Savelsbergh (1999). A computation study of search strategies for mixed integer programming. *INFORMS Journal on Computing 11*(2), 173–187.

Melo, M. T., S. Nickel, and F. Saldanha-da Gama (2008). Facility location and supply chain management – a review. *European Journal of Operational Research 196*(2), 401–412.

Mirchandani, P. and R. Jagannathan (1989). Discrete facility with nonlinear diseconomies in fixed costs. *Annals of Operations Research 18*(1–4), 213–224.

Nemhauser, G. L. and P. H. Vance (1994). Lifted cover facets of the 0–1 knapsack polytop with GUB constraints. *Operations Research Letters 16*(5), 255–263.

Nemhauser, G. L. and L. A. Wolsey (1999). *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience.

Nozick, L. K. and M. A. Turnquist (1998). Integrating inventory impacts into a fixed-charge model for locating distribution centers. *Transportation Research Part E (Logistics and Transportation Review) 34*(3), 173–186.

Nozick, L. K. and M. A. Turnquist (2001). A two-echelon inventory allocation and distribution center location analysis. *Transportation Research 37*(6), 425–441.

Pisinger, D. (1995). A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research 83*(2), 394–410.

Qi, X. (2007). Order splitting with multiple capacitated suppliers. *European Journal of Operational Research 178*(2), 421–432.

ReVelle, C. S., H. A. Eiselt, and M. S. Daskin (2008). A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research 184*(3), 817–848.

ReVelle, C. S. and G. LaPorte (1996). The plant location problem: New models and research prospects. *Operations Research 44*(6), 864–874.

Schütz, P., L. Stougie, and A. Tomasgard (2008). Stochastic facility location with general long-run costs and convex short-run costs. *Computers & Operations Research 35*(9), 2988–3000.

SCIP (2010). Solving Constraint Integer Programs. More info on project homepage `http://scip.zib.de/scip.shtml`.

Shen, Z.-J. M. (2007). Integrated supply chain design models: A survey and future research directions. *Journal of Industrial and Management Optimization 3*(1), 1–27.

Shen, Z.-J. M., C. R. Coullard, and M. S. Daskin (2003). A joint inventory-location model. *Transportation Science 37*(1), 40–55.

Sinha, P. and A. A. Zoltners (1979). The multiple-choice knapsack problem. *Operations Research 27*(3), 503–515.

Soland, R. M. (1974). Optimal facility location with concave costs. *Operations Research 22*(2), 373–382.

Sridharan, R. (1995). The capacitated plant location problem. *European Journal of Operational Research 87*(2), 203–213.

Vielma, J. P., S. Ahmed, and G. Nemhauser (2010). Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research 58*(2), 303–315.

Vielma, J. P., A. B. Keha, and G. L. Nemhauser (2008). Nonconvex, lower semicontinuous piecewise linear optimization. *Discrete Optimization 5*(2), 467–488.

Wollenweber, J. (2008). A multi-stage facility location problem with staircase costs and splitting of commodities; model, heuristic approach and application. *OR Spectrum 30*(4), 655–673.

Wolsey, L. A. (1998). *Integer Programming*. New York, NY, USA: Wiley-Interscience publication.

Wu, L.-Y., X.-S. Zhang, and J.-L. Zhang (2006). Capacitated facility location problem with general setup cost. *Computers & Operations Research 169*(5), 1226–1241.

Zangwill, W. I. (1968). Minimum concave cost flows in certain networks. *Management Science 14*(7), 429–450.

Zemel, E. (1980). The linear multiple choice knapsack problem. *Operations Research 28*(6), 1412–1423.

# DEPARTMENT OF ECONOMICS AND BUSINESS

AARHUS UNIVERSITY
SCHOOL OF BUSINESS AND SOCIAL SCIENCES
www.econ.au.dk

## PhD Theses since 1 July 2011

2011-4 Anders Bredahl Kock: Forecasting and Oracle Efficient Econometrics

2011-5 Christian Bach: The Game of Risk

2011-6 Stefan Holst Bache: Quantile Regression: Three Econometric Studies

2011:12 Bisheng Du: Essays on Advance Demand Information, Prioritization and Real Options in Inventory Management

2011:13 Christian Gormsen Schmidt: Exploring the Barriers to Globalization

2011:16 Dewi Fitriasari: Analyses of Social and Environmental Reporting as a Practice of Accountability to Stakeholders

2011:22 Sanne Hiller: Essays on International Trade and Migration: Firm Behavior, Networks and Barriers to Trade

2012-1 Johannes Tang Kristensen: From Determinants of Low Birthweight to Factor-Based Macroeconomic Forecasting

2012-2 Karina Hjortshøj Kjeldsen: Routing and Scheduling in Liner Shipping

2012-3 Soheil Abginehchi: Essays on Inventory Control in Presence of Multiple Sourcing

2012-4 Zhenjiang Qin: Essays on Heterogeneous Beliefs, Public Information, and Asset Pricing

2012-5 Lasse Frisgaard Gunnersen: Income Redistribution Policies

2012-6 Miriam Wüst: Essays on early investments in child health

2012-7 Yukai Yang: Modelling Nonlinear Vector Economic Time Series

2012-8 Lene Kjærsgaard: Empirical Essays of Active Labor Market Policy on Employment

2012-9 Henrik Nørholm: Structured Retail Products and Return Predictability

2012-10 Signe Frederiksen: Empirical Essays on Placements in Outside Home Care

| | |
|---|---|
| 2012-11 | Mateusz P. Dziubinski: Essays on Financial Econometrics and Derivatives Pricing |
| 2012-12 | Jens Riis Andersen: Option Games under Incomplete Information |
| 2012-13 | Margit Malmmose: The Role of Management Accounting in New Public Management Reforms: Implications in a Socio-Political Health Care Context |
| 2012-14 | Laurent Callot: Large Panels and High-dimensional VAR |
| 2012-15 | Christian Rix-Nielsen: Strategic Investment |
| 2013-1 | Kenneth Lykke Sørensen: Essays on Wage Determination |
| 2013-2 | Tue Rauff Lind Christensen: Network Design Problems with Piecewise Linear Cost Functions |

ISBN: 9788790117986