CHALMERS | UNIVERSITY OF GOTHENBURG

*MASTER'S THESIS*

# Minimizing costs for transport buyers using integer programming and column generation

Eser Esirgen

*Department of Mathematical Sciences*
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Göteborg, Sweden 2010

Thesis for the Degree of Master of Science

# Minimizing costs for transport buyers using integer programming and column generation

Eser Esirgen

CHALMERS | UNIVERSITY OF GOTHENBURG

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg, Sweden
Göteborg, July 2010

Mathematical Sciences
Göteborg 2010

**Abstract**

Proxio Optimizer is a software developed by Proxio AB and designed for transportation buyers who want to reduce cost and environmental impact. During the development phase of the software, mathematical optimization techniques have, however, never been considered as a key tool. Hence, developing a mathematical optimization model in order to evaluate the performance and efficiency of the software developed by Proxio is the aim of this project.

The problem to be solved is modeled by two optimization models. The first model is called the column generation model; it is utilized to generate paths for transporting goods from the origins to the respective destinations. The second model is called the main optimization model; it uses the generated paths and optimizes the total cost over these paths.

The results from tests of the mathematical model show that Proxio Optimizer is an efficient optimization engine, since the results from the optimization model and Proxio Optimizer are very similar. Additionally, the column generation model appears to be a very effective tool for decreasing the size of the mathematical model, which in turn reduces the computation time required to solve the problem.

**Acknowledgements**

This project was carried out in cooperation with Proxio from January to June 2010. First, I would like to express my deepest appreciation to my examiner Ann-Brith Strömberg for her contribution to extend my knowledge and being an excellent tutor throughout this research. Besides, I would like to thank Mattias Bergström from Proxio, who provided me the chance to conduct this project at Proxio. I would like to thank my dearest friend Cagri Tilki for his continuous support to me since high school. Finally, I would like to express my warmest thanks to my beloved parents for their unconditional love and support during all my life and during my studies in Chalmers University of Technology.

Göteborg, June 2010
Eser Esirgen

# Contents

# 1 Introduction

To be successful in today's highly competitive marketplaces, companies must strive for greatest efficiency in all of their activities and completely utilize any possible opportunity to gain a competitive advantage over other firms. Among many possible activities, cost reduction in logistics is regarded as one of the core areas presenting enormous opportunities.

Jonsson [1] separates the costs of logistics into two: direct and indirect costs. Direct costs include physical handling, transportation, and storage of goods in the flow of materials together with the administration costs, whereas capacity and shortage costs are indirect costs. He further asserts that direct logistics costs roughly vary between 10% and 30% of the turnover depending on the type of industry [1]. This implies that the costs of logistics hold a large portion among all possible expenses for companies. That is why any company being able to reduce this huge portion, will definitely obtain a significant decrease in their total expenses.

In order to be able to reduce the portion of costs originating from logistics, one should start by understanding and investigating the logistics management and environment. G. Stefansson defines logistics management to be the process of planning, implementing and controlling the efficient, effective flow and storage of raw materials, in-process inventory, finished goods, services, and related information from the point of origin to the point of consumption [3].

As the definition above points out, logistics management is initialized by the planning period and is followed by implementation and monitoring phases. Thus, a successful implementation of a logistics system is highly correlated to how good logistics management plans are. In accordance with this, Dov, Tzvi and Shenhar explain that a minimum level of planning is essential for success and further assert that planning is a core subject for prospering implementation [4].

In such an environment, it is believed that implementing optimization techniques to transportation of goods in order to schedule when and how much to send from each origin origin to its respective destination over a certain time period is a possible way to make improvements over the total cost of logistics.

## 1.1 About Proxio

This thesis has been conducted at the company Proxio AB, which is located in Mölnlycke, outside Gothenburg [2]. Proxio has developed a software called Proxio Optimizer to find solutions to the logistics problems of their customers. Their customer portfolio includes companies from different industries.

Proxio Optimizer is a powerful, yet easy-to-use, software designed for transportation buyers who want to reduce costs and environmental impact. Proxio first helps the user to calculate, analyze and visualize the costs and environmental footprints of the current transport setups. It then proposes a number of changes in the transport setup, e.g. which carriers to use and how often to use them, in order to significantly reduce both cost and emissions. Proxio performs these calculations by utilizing a heuristic method, defined and implemented in the software by the company. Based on the results of the Proxio Optimizer, a full fledged implementation typically leads to a reduction in transportation cost by 10% to 30% and a significant decrease in environmental impact [2].

However, during the development phase of this software, mathematical optimization techniques has never been considered as a key tool. That is why the aim of the project, which is made on behalf of Proxio, is to develop a mathematical optimization model in order to evaluate the solutions computed by the software developed by Proxio.

## 1.2 Aim of the Project

The goal of this thesis is to develop a mathematical optimization model, to implement this model into the mathematical modeling software AMPL [10] and to solve it by the optimization solver CPLEX [11]. The aim is that the mathematical model should find either the optimal solution or a solution that is at least as good as the one computed by Proxio Optimizer along with lower bounds on the optimal solution for the instances provided by the company. Furthermore, findings will be used to compare the results of the Proxio Optimizer to the results of the optimization program that will be developed. As a result, it will be possible to evaluate the solutions computed by the Proxio Optimizer.

The environmental impact analysis part of the Proxio Optimizer is not studied in this project, since the company is largely interested in how effective their software is, considering reduction of the total logistics costs. However, the environmental objective can be modeled similar to the economical one studied in this work.

## 1.3 Methodology

In this project, several important tasks have been performed: from understanding and defining the problem to constructing a mathematical model, decomposing the model for solvability, performing relevant tests, and analyzing results.

The project was initialized by experimenting with the Proxio Optimizer. This part was performed so as to observe and understand the capabilities of the software. After that, several meetings were held with the employee of the company in order to clarify the problem. At this stage, it was decided what to include and what not to include in the mathematical model that should be developed. Having decided on the scope of the program, several weeks were spent on developing the model. The development phase was conducted iteratively between meetings with supervisors at Chalmers and at the company so that the final model would be not only accurate in terms of optimization modeling and solution techniques but also fulfilling the expectations of the company. During this phase, it was obvious that the mathematical model would be huge and computationally very hard to solve. Therefore, it was decided that several optimization techniques to decompose the model should be utilized.

Having developed an acceptable model, the second part of the project was initialized which was to implement the model into the mathematical modeling software called AMPL for solving by the optimization solver CPLEX. The very first attempts to solve the mathematical model by including all the variables were futile, since AMPL could not even load the huge model. Afterwards, several decomposing techniques were utilized and the number of optimization variables was drastically decreased, while retaining variables having large potentials for producing good solutions. As a result, CPLEX managed to provide useful results for analysis and discussions by using the restricted model.

## 1.4 Outline

In the next section, a comprehensive description of the problem is given together with some illustrations. After that, several topics in mathematical optimization theory which are related to the problem are introduced. In Section 4, the optimization model together with its extensive definitions are introduced. In Section 5, test instances together with their results are given and analyzed. Besides these, this section includes comparisons of the results from Proxio Optimizer and the mathematical optimization model. In the final section, further developments of Proxio Optimizer and the mathematical model are suggested.

# 2 The Problem

Since the optimization program that will be developed is expected to be applicable to different instances, this section starts with depicting the scope of the problem which is followed by an extended description of the problem through a case provided by the company.

## 2.1 Definition of the Problem

As discussed in Section 1, logistics management involves many different activities such as flow, storage, and handling of goods. In order to be able to define a possibly solvable problem out of such a huge system, some activities must be selected to be the core of this study, while others are neglected.

Among many activities that are possible to plan by utilizing mathematical optimization techniques, effective flows together with several decisions related to this activity are considered to be the problem focus. These decisions consist of

- selecting a transportation path for each origin-destination pair,
- the weekly frequency of each selected path,
- the departure days of the goods from their points of origins, and
- the amount of the goods to be transported in each departure from its respective origin.

In Proxio Optimizer, each of these decisions is constrained by several rules. These rules are as follows:

- Among all possible paths, there should be only one active path for each origin-destination pair, and this path is the same throughout planning period.
- Each path is assigned a weekly frequency. Once the weekly frequency of each path is fixed, then it is the same throughout the planning period. That is, if the frequency of a path is zero for *Week 1*, then the frequency of that path is zero throughout the planning period.
- The number of departure days from the origin of a path is equal to the frequency of that path, and this number is constant over all the weeks of the planning period. Moreover, the departure days are the same for all different weeks. That is, if *Monday* and *Thursday* are selected to be the departure days of a path on *Week 1*, then these days are the departure days for all different weeks of the planning period.
- The Proxio Optimizer is developed to distribute the weekly amount for each origin-destination pair equally among the departure days. That is, if the frequency of path $k$ which connects a certain origin-destination pair, is two, then half of the weekly demand for this origin-destination pair is transported at each departure through that path.

In light of these rules, the problem is to find such a logistics network that the total cost of the transportation system is minimized.

## 2.2 Description of the Problem

Even though the problem itself involves many inter-relations between the decision points, the problem contains four major steps, as displayed in Figure 1, from selecting a path for each origin-destination pair to deciding how much of the goods to be transported at each departure.

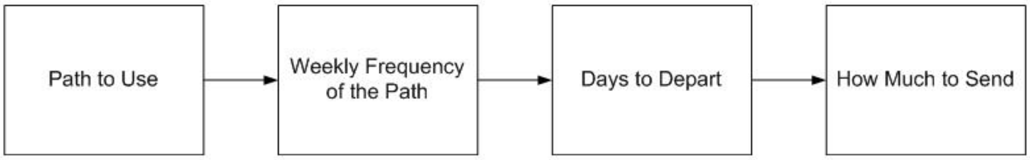


Figure 1: Problem flow

What makes these steps major is associated with how the problem is defined by Proxio. The problem is initialized when the customer company decides to plan its inbound flow of goods. This means that the company expects to know in advance, when and how much of a certain freight will be received. Depending on this, several input data is defined.

- *Planning period*: Since plans are established over a certain period of time, it should be pre-determined. In the case studied in this work, the planning period is 26 weeks.
- *Weekly flows from each origin point to its destination*: As the planning period is defined to be 26 weeks, the company should estimate how much to produce during this period. Since production requires materials, the company estimates how much of each material is demanded using statistics of the estimated production. Therefore, the company forms a list, containing from where and how much to convey. The problem instance provided by the company, includes 31 origin-destination pairs and the weekly amount of flow of goods between each of these pairs. This is visualized in Figure 2.

| Week | Origin Location | Destination Location | Weight [kg] |
|---|---|---|---|
| 1 | A | D | 1000 |
| 2 | A | D | 3000 |
| | ........................................ | | |
| 26 | A | D | 2500 |
| 1 | B | D | 7000 |
| | ........................................ | | |
| 26 | B | D | 3000 |
| | ........................................ | | |

Figure 2: Origin-destination pairs and corresponding demands for the respective weeks in the planning period

- *Hubs*: Hubs are defined to be the cross-docking points where several goods coming from different locations can be consolidated as long as their next destinations are the same. Hubs are essential for the reduction of transportation

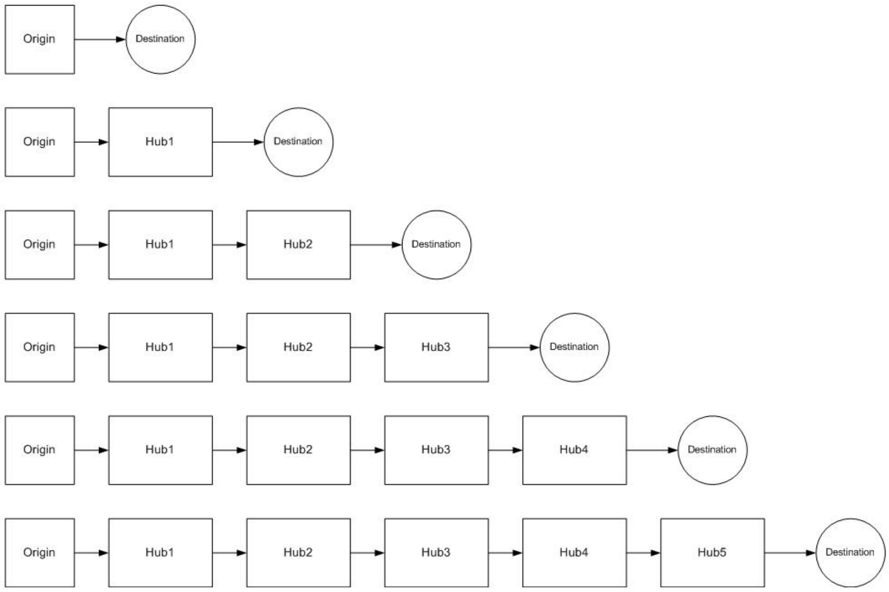


Figure 3: Examples of possible paths from an origin to a destination

costs, since cost per kg decreases as the magnitude of freight increases; see the description of cost profiles below. Six different hubs appear in this study.

- *Paths*: Paths are the connections between each origin-destination pair. Each path can include up to five hubs. Figure 3 exemplifies several possible paths for an origin-destination pair.

  Since the problem has 31 origin-destination pairs together with several possible hubs, the system contains a vast number of paths in total. The number of paths including $k$ hubs is thus,

  $$31 \cdot \frac{6!}{(6-k)!},$$

  which yields a total of

  $$31 \cdot \sum_{k=0}^{5} \frac{6!}{(6-k)!} = 38317$$

  possible paths, which should also be multiplied by the number of days in the planning period.

- *Cost Profiles*: Every path consists of an origin-destination pair together with a possibility to visit several hubs. Therefore, each path consists of several links between cities. In accordance with this link-path structure, transportation costs are calculated with respect to the total amount of goods conveyed through each of the links between these stations; these cost profiles are different for each link. Figure 4 shows an example of the cost profile for the link between *Hub1* and *Hub2* as a function of the amount of goods sent over the link.

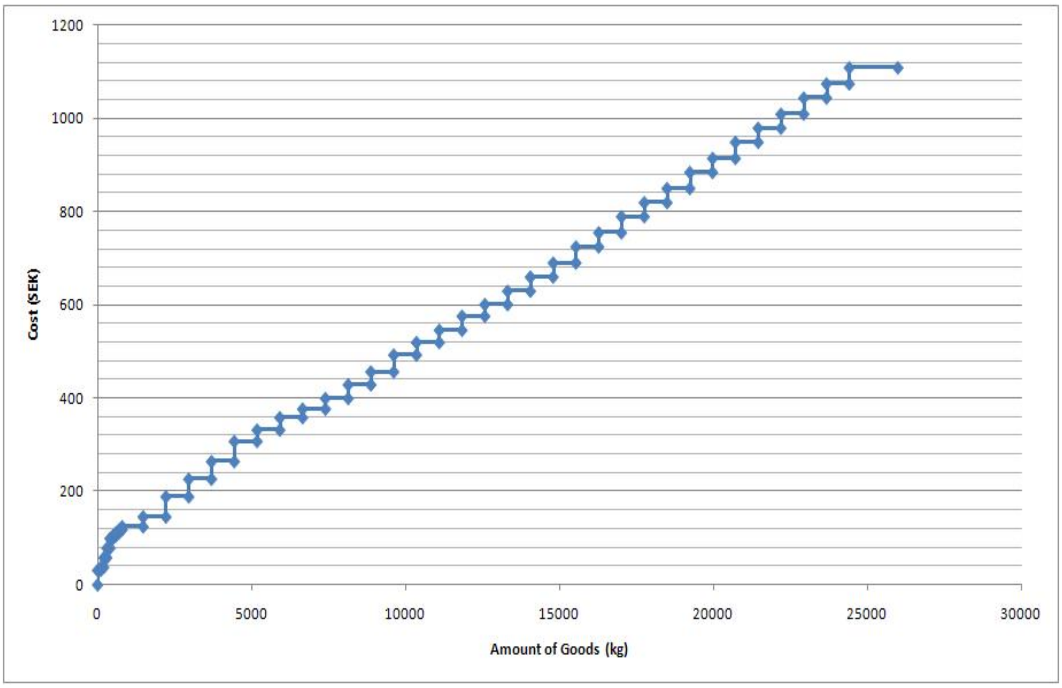


Figure 4: Cost profile of the link between Hub1 and Hub2

As pointed out in Figure 4, the cost profile function of the link between *Hub1* and *Hub2* reaches its highest value at around 25 tons and is defined up to 26 tons, which defines the capacity of trucks on this link. If the magnitude of goods is greater than 26 tons, more space has to be purchased from additional trucks using the same cost function.

One other significant point about the cost function is that there is a fixed cost for each interval. That is, if there exists an interval between 1000kg and 3000kg with a fixed cost of 500SEK, then transporting 1000kg or 2999kg costs the same, which is 500SEK.

Having introduced all the data, the problem is to define one path for each origin-destination pair so that the total cost of the transportation system will be minimized while fulfilling all weekly demands. Additionally, while fulfilling the demands for each origin-destination pair over a certain path, some days, which will be the same every week throughout the planning period, have to be identified as the departure days for each week and the demands should be evenly distributed among these days. But, an even distribution of the demands among departure days will not be included in optimization models, since it leads to non-linear relations which are out of the scope of the type of mathematical models to be developed in this study.

# 3 Theory

This section contains a comprehensive summary of some of the theory within the area of mathematical optimization. The theory that is introduced, is required to develop an efficient model for the problem described in the previous section. These are linear programming, mixed integer linear programming, linear network flow models, and column generation.

## 3.1 Linear Programming

Linear programming (LP) constitutes an essential part of this master thesis, since many efficient algorithms exist to solve linear programming models. Thus, understanding the concepts of linear programming is very beneficial. Below is a short introduction to linear programming. For an extended text on linear programming, see e.g. [5] and [12].

The word linear indicates that linear programming problems are restrained by linear constraints (i.e., equalities and/or inequalities), and the quality of the solution is measured by a linear function of the considered quantities. A very simple example of a linear programming model is given by

$$\begin{aligned} \text{minimize} \quad z = \ & x_1 + x_2, \\ \text{subject to} \quad & x_1 + x_2 \geq 5, \\ & x_1 - 2x_2 \leq 2, \\ & x_1, x_2 \geq 0. \end{aligned}$$

A linear program is often expressed by matrices and vectors as a system of inequalities and/or equations. Considering that minimizing an objective function $\mathbf{c}^T\mathbf{x}$ is equivalent to maximizing $-\mathbf{c}^T\mathbf{x}$, each linear program can be modified and expressed as to

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x}, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}^n, \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a real matrix, $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ are given vectors, and $\mathbf{0}^n$ denotes a vector with $n$ elements of 0.

Among all vectors $\mathbf{x} \in \mathbb{R}^n$ that satisfy a given system of linear equations and inequalities, finding a vector $\mathbf{x}^* \in \mathbb{R}^n$ that minimizes (or maximizes) the value of a given linear function is the general aim of a linear program. The linear function that is to be either minimized or maximized is called the *objective function*. The linear equations and inequalities in the linear program which define the feasible set of the problem are called the *constraints*.

Any vector $\mathbf{x} \in \mathbb{R}^n$ is called a *feasible solution*, as long as it satisfies all the constraints of a given linear program. A solution $\mathbf{x}^*$ satisfying the constraints is called the *optimal solution*, if $\mathbf{c}^T\mathbf{x}^* \leq \mathbf{c}^T\mathbf{x}$ for all $\mathbf{x} \geq \mathbf{0}^n$ such that $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Despite having a single optimal solution in general, a linear program may have infinitely many solutions or none at all.

- If an optimal solution exists to a linear program, at least one optimal solution is located in an extreme point to the feasible set.
- If all points on the edge of the polyhedron defined by the constraints, $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq 0$, are optimal, then the linear program has infinitely many solutions.
- If a linear program lacks feasible solutions, then there is no optimal solution to that problem. Such a linear program is called *infeasible*.
- Some linear programs with feasible solutions, might lack an optimal solution. This happens, when the objective function can attain arbitrarily large values. Such linear programs are called *unbounded*.

There are several different methods to solve linear programs. Among all, the simplex method and interior point methods are the most famous ones. The simplex method initializes by constructing a feasible solution at a vertex of the polyhedron, and continues iteratively by visiting other vertices with non-increasing values of the objective function until an optimum is reached. Contrary to the simplex method, the interior point method is initialized at a point in the interior of the feasible region, and reaches the optimum by searching inside this region rather than visiting the vertices. Moreover, it is shown that the average number of iterations for the simplex algorithm is bounded by $\mathcal{O}(\min\{(m-n)^2, n^2\})$ where $m$ is the number of constraints and $n$ is the number of variables, while the interior point algorithm can find the optimum in polynomial-time [12].

## 3.2 Mixed Integer Linear Programming

Having introduced the basic theory behind linear programming, its generalization to Mixed Integer Linear Programming (MILP) requires further description. MILP constitutes an important part of this master thesis, because of several restrictions presented in the problem. Selecting a single path for each origin destination pair, defining a weekly transport frequency for that path together with defining possible departure days, require the use of either binary or integer variables. On the other hand, the variables representing the amount to be transported at each departure do not require integrality.

These all together point out that several integral variables have to be utilized together with some continuous ones. In such a case, the theory behind MILP becomes very important in order to develop a model that can be efficiently solved. Therefore, understanding the concept of MILP is valuable. Below is a short introduction to MILP; for an extended description on MILP, see e.g. [6].

A MILP appears to be very similar to the linear program introduced in Section 3.1 except for the restriction that certain variables must take integer values.

Taking this exception into account, a general formulation of the MILP looks as follows:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} + \mathbf{h}^T\mathbf{y}, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{G}\mathbf{y} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}^n, \\ & \mathbf{y} \geq \mathbf{0}^p \text{ and integer}, \end{aligned} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{G} \in \mathbb{R}^{m\times p}$ are given real matrices, and $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{h} \in \mathbb{R}^p$ and $\mathbf{b} \in \mathbb{R}^m$ are given vectors of parameters.

If the MILP (2) involves only integer variables, then it is called an Integer Problem (IP), and no one to date has found an efficient (polynomial) algorithm for general integer problems [6]. It means that the complexity and the computational time of the IP's are much higher than that of an LP. Taking this difficulty into account, it it clear that as the number of the integer and the binary variables increase in a MILP model, the time it takes to solve a MILP to optimality is much longer than that of a LP (at least higher than polynomial time). In some cases, it is impossible to solve a MILP model to optimality, while this case is very rare for LP models.

It is pointed out above that the majority of the variables are expected to require integrality in the optimization problem introduced in Section 2.2. Therefore, the optimization model to be solved in this thesis is expected to possess huge computational times unless a reduction in the integer variables is performed.

## 3.3 Network Flow Models

A road network that can be used for transporting goods from a source to a sink can be modeled as a network. In the network, the cities are represented by nodes, and each link represents a section of a road that connects a city to another. An example network is shown in Figure 5. The link capacity is defined to be the maximum amount of goods that can be conveyed through the corresponding arc (e.g., per unit of time). Similarly, a network could as well represent a pipeline network for transporting oil or a network of phone lines.

The instance provided by Proxio shows many similarities with the network flow structure. The problem consists of many arcs which are used to form the paths between origin-destination pairs. Additionally, the cost functions introduced in Section 2.2 are defined for each arc in the problem. Therefore, the theory of network flow models is an important part of this study. Below is a short introduction to the mathematical modeling and solution of problems involving flows in networks. For further reading, see e.g. [9].

The example in Figure 5 is defined by the following:

- $N = \{s,1,2,3,4,t\}$ is the set of nodes.
- $A = \{(s,1),(s,3),(1,2),(2,4),(2,t),(3,1),(3,4),(4,t)\}$ is the set of directed arcs.

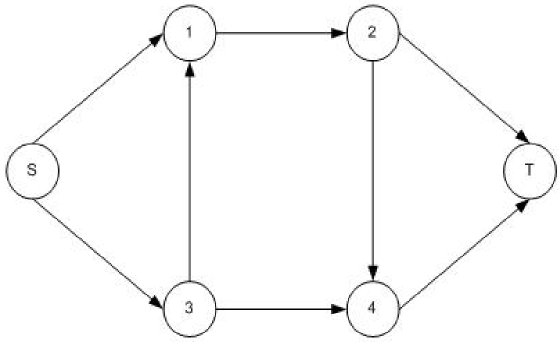


Figure 5: An example of a network with nodes and arcs connecting the nodes

- If $s$ and $t$ denote the source and the sink nodes, respectively ,of an origin destination pair then, then $P = \{(s,1,2,t),(s,3,4,t),(s,1,2,4,t),(s,3,1,2,t),(s,3,1,2,4,t)\}$ is the set of paths between this pair.

Assume that $x_{ij}$ and $c_{ij}$ denote the amount of flow and the capacity, respectively, of arc $(i,j)$. Then, any flow defined by $\{x_{ij} \mid (i,j) \in A\}$, and satisfying the constraints

$$0 \le x_{ij} \le c_{ij}, \quad (i,j) \in A,$$

$$\sum_{j \in N:(j,i) \in A} x_{ji} - \sum_{j \in N:(i,j) \in A} x_{ij} = \begin{cases} -v, & i = s, \\ 0, & i \in N \backslash \{s,t\}, \\ v, & i = t, \end{cases}$$

is called a feasible flow from $s$ to $t$ of the total amount $v$, where $s$ and $t$ are the source and the sink nodes, respectively.

## 3.4 Column Generation and the Multicommodity Minimal Cost Flow Problem

The total number of paths calculated in Section 2.2 shows that a vast model has to be solved in order to find a feasible solution. In addition to that, the cost profiles of links introduced in the same section are in a piece-wise linear form. Such a form, can be handled in a MILP only by utilizing binary variables. All these requirements are expected to cause the program's computation time to increase drastically. Moreover, it has been explained in Section 3.2 that MILPs are much harder to solve than LPs. Therefore, developing a mathematical model that reduces the number of binary variables before solving the main model is a must. For such a case, it is believed that the *column generation* method is a good choice.

Having described the requirement to the column generation method and the network flow modeling briefly, it is believed that introducing a general problem called the *Multicommodity Minimal Cost Flow Problem* (MMCFP), which is somewhat similar to the problem that is focused in this work, is highly beneficial. Below is a short introduction to the (MMCFP), followed by the application of column generation to this problem. For further reading on column generation, see e.g. [8]. For further reading on the *Multicommodity Minimal Cost Flow Problem*, see e.g. [7].

MMCFP consists of $t$ different commodities flowing in a network consisting of $m$ nodes and $n$ arcs. Each commodity has its own upper flow limit on each arc, while the system itself has an upper bound on the sum of the flow of all the commodities. If the vector of upper limits for commodity $i$ is defined by $\mathbf{u}_i \in \mathbb{R}^n$, then $u_{ipq} \in \mathbb{R}$ represents the upper limit for the flow of commodity $i$ through the arc $(p,q)$. Similarly, $u_{pq} \in \mathbb{R}$ defines the upper limit of the total flow through the arc $(p,q)$, while $\mathbf{u} \in \mathbb{R}^n$ is the upper bound for the total flow in the system. Furthermore, if the cost of flow of commodity $i$ in arc $(p,q)$ is $c_{ipq} \in \mathbb{R}$, then the vector $\mathbf{c}_i \in \mathbb{R}^n$ represents the cost vector of commodity $i$. If the vector of demands for commodity $i$ is defined by the vector $\mathbf{b}_i \in \mathbb{N}^m$, then the MMCFP is formulated as to

$$
\begin{aligned}
g^* = \text{minimize} \quad & \sum_{i=1}^{t} \mathbf{c}_i \mathbf{x}_i, \\
\text{subject to} \quad & \sum_{i=1}^{t} \mathbf{x}_i \leq \mathbf{u}, \\
& \mathbf{A}\mathbf{x}_i = \mathbf{b}_i, \qquad i = 1, \ldots, t, \\
& \mathbf{0} \leq \mathbf{x}_i \leq \mathbf{u}_i, \qquad i = 1, \ldots, t,
\end{aligned}
\tag{3}
$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the node-arc incidence matrix and $\mathbf{x}_i \in \mathbb{R}^n$ is the vector representing the flows of commodity $i$ in the network. The rows of the node-arc incidence matrix represent the nodes and its columns represent the arcs. It assigns 1 (-1) to an element if the corresponding node is the beginning (end) of the corresponding arc. All other elements possess the value 0. The node-arc incidence matrix corresponding to the network in Figure 5 is given by

$$
\begin{array}{c}
\\ s \\ 1 \\ 2 \\ 3 \\ 4 \\ t
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}
(s,1) & (s,3) & (1,2) & (2,4) & (2,t) & (3,1) & (3,4) & (4,t)
\end{array} \\
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -1
\end{pmatrix}
\end{array}
$$

Having introduced the MMCFP model (3), an application of the column generation principle to this problem is given by the following.

The equality constraint, $\mathbf{A}\mathbf{x}_i = \mathbf{b}_i$, and the upper and lower limits on $\mathbf{x}_i$ given by $\mathbf{0} \leq \mathbf{x}_i \leq \mathbf{u}_i$ where $\mathbf{u}_i$ is assumed to be bounded, define the convex set,

$$X_i = \{\mathbf{x}_i : \mathbf{A}\mathbf{x}_i = \mathbf{b}, \mathbf{0} \leq \mathbf{x}_i \leq \mathbf{u}_i\}, \quad i = 1, \ldots, t.$$

Let $\mathbf{x}_{ij}, j = 1, \ldots, k_i$, denote the extreme points of the set $X_i$. Then, convex combinations of these extreme points can be used to express any $\mathbf{x}_i \in X_i$ according to

$$\mathbf{x}_i = \sum_{j=1}^{k_i} \lambda_{ij}\mathbf{x}_{ij}, \tag{4}$$

where

$$\sum_{j=1}^{k_i} \lambda_{ij} = 1, \quad \lambda_{ij} \geq 0, \quad j = 1, \ldots, k_i. \tag{5}$$

If the slackness vector is denoted by $\mathbf{s}$ and $\mathbf{x}_i$ is replaced by the convex combinations defined by (4)-(5), then MMCFP (3) can be reformulated as the following *master problem*, which is defined over the arcs and the paths of the network, to

$$\begin{aligned}
g^* = \text{minimum} \quad & \sum_{i=1}^{t}\sum_{j=1}^{k_i} (\mathbf{c}_i^T \mathbf{x}_{ij})\lambda_{ij}, \\
\text{subject to} \quad & \sum_{i=1}^{t}\sum_{j=1}^{k_i} \mathbf{x}_{ij}\lambda_{ij} + \mathbf{s} = \mathbf{u}, \\
& \sum_{j=1}^{k_i} \lambda_{ij} = 1, \quad i = 1, \ldots, t, \\
& \lambda_{ij} \geq 0, \quad j = 1, \ldots, k_i, \quad i = 1, \ldots, t, \\
& \mathbf{s} \geq \mathbf{0}.
\end{aligned} \tag{6}$$

In the master problem (6), the number of variables ($\lambda_{i,j}$) is huge and they increase as the number of possible paths increase. Therefore, the following *restricted master problem* (7), according to $\overline{g}^* \geq g^*$, is formulated to have much fewer variables by considerably decreasing the number of possible paths, but still taking feasibility into consideration:

$$\overline{g}^* = \text{minimize} \quad \sum_{i=1}^{t}\sum_{j=1}^{k_i} (\mathbf{c}_i^T \mathbf{x}_{ij})\lambda_{ij}, \tag{7a}$$

$$\text{subject to} \quad \sum_{i=1}^{t}\sum_{j=1}^{k_i} \mathbf{x}_{ij}\lambda_{ij} + \mathbf{s} = \mathbf{u}, \tag{7b}$$

$$\sum_{j=1}^{k_i} \lambda_{ij} = 1, \quad i = 1, \ldots, t, \tag{7c}$$

$$\lambda_{ij} \geq 0, \quad j = 1, \ldots, \overline{k_i}, \quad i = 1, \ldots, t, \tag{7d}$$

$$\mathbf{s} \geq \mathbf{0}, \tag{7e}$$

where $1 \leq \overline{k_i} \leq k_i$, and $i = 1, \ldots, t$.

Initializing the restricted master problem (7) with a basic feasible solution, new and promising paths (that will yield a lower objective value of the problem (7)) are generated by solving a subproblem to the restricted master problem. The subproblem, which separates over the commodities $i = 1, \ldots, t$, and is defined over the arcs of the network, is to

$$\begin{aligned} h_i(\mathbf{w},\alpha_i) = \text{minimize} \quad & (\mathbf{c}_i - \mathbf{w})^T \mathbf{x}_i - \alpha_i, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x}_i = \mathbf{b}_i, \\ & \mathbf{0} \leq \mathbf{x}_i \leq \mathbf{u}_i, \end{aligned} \tag{8}$$

where $\mathbf{w}$ and $\alpha_i$ denote the (vector of) dual variables corresponding to the first (7b) and the second (7c) constraint group in the restricted master problem (7) in an optimal solution of this problem. The generation of new columns and solution of the resulting restricted master problem work iteratively until no new paths are generated or the total costs of all subproblems (8) are positive which imply that there is no possibility to decrease the total cost of the restricted master problem (7).

The master problem (6) is generated from the MMCFP (3) and they have the same optimal value, while the restricted master problem (7) is generated from the master problem (6) by utilizing (possibly) fewer paths. Hence, the optimal value of the restricted master problem (7) is an upper bound to the master problem (6) and the MMCFP (3), i.e., $\overline{g}^* \geq g^*$. Moreover, summing up the total costs calculated by the subproblem (8) over each commodity, i.e., $h^*(\mathbf{w},\alpha) = \sum_{i=1}^{t} h_i(\mathbf{w},\alpha_i)$, gives a lower bound to the master problem (6) and the MMCFP (3), i.e., $h^*(\mathbf{w},\alpha) \leq g^*$.

The MMCFP (3), the master problem (6) and the restricted master problem (7) are continuous problems. On the contrary, the comprehensive description of the optimization problem in Section 2.2 points out that there should be integer variables in the model because of the piece-wise linear cost functions for each link and the restriction of selecting one path for each origin-destination pair. Since integrality requirements cause a linear minimization problem's total cost to increase, and the restricted master problem (7) is the continuous (LP) relaxation of the optimization problem at hand, an optimal solution to the restricted master problem (7) yields a lower bound on the optimal value of the optimization problem to be solved.

To summarize, the column generation algorithm produces the bounds $h^*(\mathbf{w},\alpha) \leq g^* \leq \overline{g}^*$ on the optimal value $g^*$ of the program 3. If the "true" optimization model has integral values, then $g^*$ is a lower bound on its optimal value.

# 4 Optimization Model

In Section 2, the logistics problem has been introduced and described, while Section 3 has been devoted to the theories applicable for modeling and solving the problem. Utilizing the information from these two sections, a column generation algorithm, which aims to find ”potentially better and promising” paths, and a main optimization program, a so-called master program, which uses the paths generated through the column generation phase, to find the global optimum, are developed. This section is devoted to describing these models thoroughly.

## 4.1 Main Model

The properties of the logistics system can be modeled as a system of linear equalities and inequalities together with a piece-wise linear cost function. It is believed that introducing the sets, the parameters, and the variables in the model just before describing the objective function and the constraints will be helpful for the reader to understand the model.

### 4.1.1 Sets

Depending on the description of the problem, the origin locations, the destination locations, the origin-destination pairs, the hubs, the arcs, all the paths that are generated through column generation phase, the planning period in weeks, and the days of each week define the sets in this problem.

- The set $O = \{1,\dots,I\}$ contains the origin locations of goods to be delivered.
- The set $D = \{1,\dots,J\}$ contains the destination locations of goods.
- The set OrgDes $= \{(o_{i_1},d_{i_1}),\dots,(o_{i_k},d_{i_k})\} \subseteq O \times D$, where $k$ is the total number of origin-destination pairs, includes all the origin-destination pairs.
- The set $H = \{1,\dots,M\}$ represents the hubs, in which the goods can be consolidated.
- The set $A = \{(O \cup H) \times (H \cup D)\} \setminus \{(a,a)|a \in H\}$ contains all possible directed arcs in the transport system.
- The set $K_i$ contains the indices for all paths from origin $o_i$ to its destination such that $(o_i,d_i) \in$ OrgDes.
- The set $P = \{p_{i,k} \mid k \in K_i \text{ and } (o_i,d_i) \in \text{OrgDes}\}$ contains all paths $p_{i,k}$ between origin $o_i \in O$ and its destination; $p_{i,k}$ is composed by a sequence of locations, according to $o_i - h_{i1} - h_{i2} - \cdots - d_i$ (see Figure 3).
- The set $W = \{1,\dots,L\}$ denotes the weeks of the planning period.
- The set $G = \{1,\dots,7\}$ contains week days (monday,..., sunday) denoted by the numbers 1,...,7.

### 4.1.2 Parameters

The description of the problem shows that the amount to be transported from each origin to its destination together with the cost profiles are the parameters of this problem. Since the cost functions, which are defined over one truck, are piece-wise linear, a few more parameters need to be introduced. These parameters are the number of linear pieces in the cost functions, the cost and the capacity limit of each piece of the cost functions, the capacities of the trucks, and the cost of renting a full truck.

- $\text{Weight}_{i,l}$ is the amount of goods (in kg's) to be transported in week $w_l$ for all $(o_i,d_i) \in \text{OrgDes}$ such that $l \in W$.
- $\text{Npiece}_{a,b}$ is the number of pieces in the cost function of link $(a,b) \in A$.
- $\text{Rate}_{a,b,c}$ is the cost associated with piece $c \in \{1,\ldots,\text{Npiece}_{a,b}\}$ in the cost function of link $(a,b) \in A$.
- $\text{Limit}_{a,b,c}$ is the upper capacity limit of piece $c \in \{1,\ldots,\text{Npiece}_{a,b}\}$ in the cost function of link $(a,b) \in A$.
- $\text{Capacity}_{a,b}$ is the capacity of one truck on link $(a,b) \in A$.
- $\text{Price}_{a,b}$ is the cost of renting one full truck on link $(a,b) \in A$.

### 4.1.3 Binary and Integer Variables

Considering the rules and the restrictions described in Section 2, the problem involves many "yes or no" decisions. These decisions involve the selection of a path for each origin-destination pair over all generated paths, and indicating which piece of each cost function that is utilized. Therefore, the binary variables

$$X_{i,k,l} = \begin{cases} 1, & \text{if path } k \in K_i \text{ is active (i.e., selected)} \\ & \text{on week } l \in W \text{ such that } (o_i,d_i) \in \text{OrgDes}, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\text{Step}_{a,b,[7(l-1)+g],e} = \begin{cases} 1, & \text{if piece } e \in \{1,\ldots,\text{Npiece}_{\text{a,b}}\} \text{ of the cost function of} \\ & \text{link } (a,b) \in A \text{ is active on week } l \in W, \text{ day } g \in G, \\ 0, & \text{otherwise,} \end{cases}$$

are introduced.

These variables are subject to the constraints

$$X_{i,k,l} \in \{0,1\}, \quad k \in K_i, \quad (o_i,d_i) \in \text{OrgDes}, \quad l \in W,$$

$$\text{Step}_{a,b,[7(l-1)+g],e} \in \{0,1\}, \quad (a,b) \in A, \quad l \in W, \quad g \in G, \quad e \in \{1,\ldots,\text{Npiece}_{a,b}\}.$$

As depicted in Section 2.2, each cost profile is defined over one truck. On the other hand, in order to transport all goods, some links may require more than one truck for certain days. To be able to cope with such a case, the number of trucks utilized for each link is defined to be an integer variable.

- $\text{Truck}_{a,b,[7(l-1)+g]}$ is the number of trucks used in link $(a,b) \in A$ on week $l \in W$, day $g \in G$. These variables are subject to the constrains

$$\text{Truck}_{a,b,[7(l-1)+g]} \geq 0, \quad \text{and integer}, \quad (a,b) \in A, \quad l \in W, \quad g \in G.$$

### 4.1.4 Continuous Variables

Having introduced the binary and the integer variables, the complete model still requires other variables. Since the total cost of the system is calculated depending on the transported amounts, the amount to be transported from an origin to its destination at each departure, and the amounts passing through the links of the active paths should be modeled by continuous variables.

Moreover, it has been defined in Section 2.1 that the Proxio Optimizer divides the amounts evenly between departures in a week, but it is not possible to model the equal shares by using linear relations. Hence, evenly distributed shares are out of the scope of this master thesis. Instead of the evenly distributed shares, the share of each departure in a week from an origin to its destination is modeled by a continuous variable, which actually enlarges the solution set of the problem, meaning that an optimal solution may possess a lower value.

- $\text{Share}_{i,k,[7(l-1)+g]}$ is the share of $\text{Weight}_{i,l}$ that is transported through path $k \in K_i$, where $(o_i,d_i) \in \text{OrgDes}$, on week $l \in W$, day $g \in G$.
- $\text{Amount}_{i,k,[7(l-1)+g]}$ is the amount of goods to be transported through path $k \in K_i$, where $(o_i,d_i) \in \text{OrgDes}$, on week $l \in W$, day $g \in G$.
- $Y_{a,b,[7(l-1)+g]}$ is the amount of goods transported through link $(a,b) \in A$ on week $l \in W$, day $g \in G$.

These variables are subject to the constraints

$$\text{Share}_{i,k,[7(l-1)+g]} \in [0,1], \quad k \in K_i, \quad (o_i,d_i) \in \text{OrgDes}, \quad l \in W, \quad g \in G,$$

$$\text{Amount}_{i,k,[7(l-1)+g]} \geq 0, \quad k \in K_i, \quad (o_i,d_i) \in \text{OrgDes}, \quad l \in W, \quad g \in G,$$

and

$$Y_{a,b,[7(l-1)+g]} \geq 0, \quad (a,b) \in A, \quad l \in W, \quad g \in G.$$

### 4.1.5 The Constraints and the Objective Function

It has been defined in Section 2.1 that there should be exactly one active path for each origin-destination pair. In addition to that, this path is the same throughout the planning period. These constraints are modeled according to

$$\sum_{k \in K_i} \sum_{l=1}^{L} X_{i,k,l} = L, \quad (o_i,d_i) \in \text{OrgDes}, \tag{9}$$

and

$$X_{i,k,l} = X_{i,k,(l+1)} = 0, \quad k \in K_i, \quad (o_i,d_i) \in \text{OrgDes}, \quad l \in W. \tag{10}$$

The weekly frequency of the active paths for each origin-destination pair and the departure days with respect to the weekly frequencies should be determined together with the active paths. Besides these, the weekly amounts to be transported should be distributed between the departure days. Even though it appears to be several decisions to be taken at the same time, all these decisions are inter-related. That is, if the portions of each day were known, then this information could be used to calculate the weekly frequency and the departure days. Therefore, the constraints (11) are introduced to cope with all these issues at once:

$$\sum_{g=1}^{7} \text{Share}_{i,k,[7(l-1)+g]} = \text{X}_{i,k,l}, \quad k \in K_i, \quad (o_i,d_i) \in \text{OrgDes}, \quad l \in W. \tag{11}$$

Since only one path is active for each origin destination pair, the right hand side of the constraint group (11) is zero except for the active path. When the right hand side of the equation is one, the model assigns numbers between zero and one to the variables in the left hand side in order to determine the portions assigned to the respective days in each week.

Moreover, it has been described in Section 2.1 that similar to having the same active path throughout the planning period, the portions of the active days of a certain path should be the same throughout the planning period. In order to apply this restriction to the model, the constraints

$$\text{Share}_{i,k,[7(l-1)+g]} = \text{Share}_{i,k,[7l+g]}, \quad k \in K_i, \quad (o_i,d_i) \in \text{OrgDes}, \quad l \in W, \quad g \in G, \tag{12}$$

are introduced. The portion transported on a certain path on each day is utilized to calculate the actual amounts of goods being transported, according to

$$\text{Weight}_{i,l} \cdot \text{Share}_{i,k,[7(l-1)+g]} = \text{Amount}_{i,k,[7(l-1)+g]}, \tag{13}$$

$$k \in K_i, \quad l \in W, \quad (o_i,d_i) \in \text{OrgDes}, \quad g \in G.$$

Even though the active paths are determined and the amounts that are transported are distributed between the days, there is still one major problem to face. The costs are calculated over the links and not over the paths. Therefore, the amounts of goods transported on the paths should be transferred to the links while keeping one important rule in mind: If several goods coming from different locations hit a node at the same day, they must be consolidated as long as the following node in their respective paths are the same. This actually is the heart of the optimization model, since the more goods are transported on the same link on the same day the less the cost per kg of the transported goods becomes.

Letting $\alpha_{a,k}$ denote the number of days required to reach node $a$ using path $k \in K_i$ such that $(o_i,d_i) \in \text{OrgDes}$, this rule is modeled by the constraints

$$Y_{a,b,[7(l-1)+g]} = \sum_{(o_i,d_i)\,\in\,\mathrm{OrgDes}} \; \sum_{k\in K_i:\;(a,b)\,\in\,p_{i,k}} \mathrm{Amount}_{i,k,[7(l-1)+g-\alpha_{a,k}]}, \tag{14}$$

$$(a,b) \in p_{i,k}, \quad l \in W, \quad g \in G, \quad \text{such that } [7(l-1)+g-\alpha_{a,k}] > 0.$$

Having calculated the amounts on each link on a certain day, this information is used to determine the active pieces of the cost functions for each link, according to the constraints

$$\sum_{e=1}^{\mathrm{Npiece}_{a,b}} \Big(\mathrm{Limit}_{a,b,e} \cdot \mathrm{Step}_{a,b,[7(l-1)+g],e} + \mathrm{Capacity}_{a,b} \cdot \mathrm{Truck}_{a,b,[7(l-1)+g]}\Big) \geq Y_{a,b,[7(l-1)+g]}, \tag{15}$$
$$(a,b) \in A, \quad l \in W, \quad g \in G,$$

and

$$\sum_{e=1}^{\mathrm{Npiece}_{a,b}} \mathrm{Step}_{a,b,[7(l-1)+g],e} \leq 1, \quad (a,b) \in A, \quad l \in W, \quad g \in G. \tag{16}$$

The constraint group (16) states that there could be at most one active piece for each link on a certain day. This is in accordance with the constraint group (15), which determines the active pieces in each link on a certain day. If the total amount transported on a certain link is above the limit of the capacity of the trucks for that link, then the variable $\mathrm{Truck}_{a,b,[7(l-1)+g]}$ becomes active, (i.e., is set to an integer larger than 1) in order to satisfy the inequality system.

The objective function of the model is formed to minimize the total cost of the transportation system. The function consists of two parts. The first part deals with the costs of the flows in the links that are below the capacity of one truck. This part is calculated by the cost functions for each link. The second part of the cost function calculates the total cost of the number of trucks that are utilized. The key issue is that this part of the cost function is zero, as long as the amount to be transported in a certain link on a certain day is below the capacity of a truck on that link. The objective is thus to

$$\text{minimize} \quad \sum_{l=1}^{L}\sum_{g=1}^{7}\sum_{(a,b)\in A}\Bigg(\sum_{e=1}^{\mathrm{Npiece}_{a,b}} \mathrm{Rate}_{a,b,e} \cdot \mathrm{Step}_{a,b,[7(l-1)+g],e} + \mathrm{Price}_{a,b} \cdot \mathrm{Truck}_{a,b,[7(l-1)+g]}\Bigg). \tag{17}$$

### 4.1.6 The Complete Model

The complete model is summarized as to

$$
\begin{aligned}
\text{minimize} \quad & \sum_{l=1}^{L}\sum_{g=1}^{7}\sum_{(a,b)\in A}\Bigg(\sum_{e=1}^{\text{Npiece}_{a,b}} \text{Rate}_{a,b,e}\cdot \text{Step}_{a,b,[7(l-1)+g],e} + \\
& \qquad \text{Price}_{a,b}\cdot \text{Truck}_{a,b,[7(l-1)+g]}\Bigg),\\
\text{subject to} \quad & \sum_{k\in K_i}\sum_{l=1}^{L} X_{i,k,l} = L, \quad (o_i,d_i)\in \text{OrgDes},\\
X_{i,k,l} &= X_{i,k,(l+1)}, \quad k\in K_i, \quad (o_i,d_i)\in \text{OrgDes}, \quad l\in W,\\
\sum_{g=1}^{7}\text{Share}_{i,k,[7(l-1)+g]} &= X_{i,k,l}, \quad k\in K_i, \quad (o_i,d_i)\in \text{OrgDes}, \quad l\in W,\\
\text{Share}_{i,k,[7(l-1)+g]} &= \text{Share}_{i,k,[7l+g]},\\
& \quad k\in K_i, \quad (o_i,d_i)\in \text{OrgDes}, \quad l\in W, \quad g\in G,\\
\text{Weight}_{i,l}\cdot \text{Share}_{i,k,[7(l-1)+g]} &= \text{Amount}_{i,k,[7(l-1)+g]},\\
& \quad k\in K_i, \quad l\in W, \quad (o_i,d_i)\in \text{OrgDes}, \quad g\in G,\\
Y_{a,b,[7(l-1)+g]} &= \sum_{(o_i,d_i)\,\in\,\text{OrgDes}}\ \sum_{k\in K_i:\,(a,b)\,\in\,p_{i,k}} \text{Amount}_{i,k,[7(l-1)+g-\alpha_{a,k}]},\\
& \quad (a,b)\in p_{i,k}, \quad l\in W, \quad g\in G,\\
& \quad \text{such that } [7(l-1)+g-\alpha_{a,k}]>0,\\
Y_{a,b,[7(l-1)+g]} &\le \sum_{e=1}^{\text{Npiece}_{a,b}}\Bigg(\text{Limit}_{a,b,e}\cdot \text{Step}_{a,b,[7(l-1)+g],e} + \\
& \qquad \text{Capacity}_{a,b}\cdot \text{Truck}_{a,b,[7(l-1)+g]}\Bigg),\\
& \qquad (a,b)\in A, \quad l\in W, \quad g\in G,\\
\sum_{e=1}^{\text{Npiece}_{a,b}} \text{Step}_{a,b,[7(l-1)+g],e} &\le 1, \quad (a,b)\in A, \quad l\in W, \quad g\in G,\\
X_{i,k,l} &\in \{0,1\}, \quad k\in K_i, \quad (o_i,d_i)\in \text{OrgDes}, \quad l\in W,\\
\text{Share}_{i,k,[7(l-1)+g]} &\in [0,1], \quad k\in K_i, \quad (o_i,d_i)\in \text{OrgDes}, \quad l\in W, \quad g\in G,\\
\text{Amount}_{i,k,[7(l-1)+g]} &\ge 0, \quad k\in K_i, \quad (o_i,d_i)\in \text{OrgDes}, \quad l\in W, \quad g\in G,\\
Y_{a,b,[7(l-1)+g]} &\ge 0, \quad (a,b)\in A, \quad l\in W, \quad g\in G,\\
\text{Step}_{a,b,[7(l-1)+g],e} &\in \{0,1\}, \quad (a,b)\in A, \quad l\in W, \quad g\in G,\\
& \qquad e\in\{1,\ldots,\text{Npiece}_{a,b}\},\\
\text{Truck}_{a,b,[7(l-1)+g]} &\ge 0, \quad \text{and integer}, \quad (a,b)\in A, \quad l\in W, \quad g\in G.
\end{aligned}
$$

## 4.2 Column Generation Model

This section includes the column generation model of the transport system considered and consists of two main parts, which works iteratively. The first part, called the *Master Problem*, runs over the generated paths (i.e., columns) and provides values to the second part, called the *Sub-Problem*, where the paths are actually generated. Then, these generated paths are added to the set of paths that have already been generated, and the *Master Problem* runs again over the extended set of paths. This procedure continues, either until no new paths are generated by the *Sub-Problem*, or until the iterations are stopped.

The values that are provided to the second part are optimal values of dual variables corresponding to constraint groups (19)-(20) in the *Master Problem*. The first group of values, the dual variables corresponding to each link in the model, are retrieved from the constraint group (19), while the second group of values, the dual variables with respect to each origin destination point in the model, are retrieved from the constraint group (20). These values are utilized to update the cost function in the *Sub-Problem* and the *Sub-Problem* phase runs over each origin-destination pair separately.

Additionally, since the column generation method is defined for linear programs and works iteratively, the cost functions are approximated by linear functions, so that the *Master Problem* and *Sub-Problem* become linear and the total computation time decrease. The linearization is performed by associating two linear functions with each piece-wise linear cost function and defining a break-even point between the linear functions so that the sum of the squares of the approximation errors made is minimized. Figure 6 shows an example of how the cost functions are approximated by two linear functions.

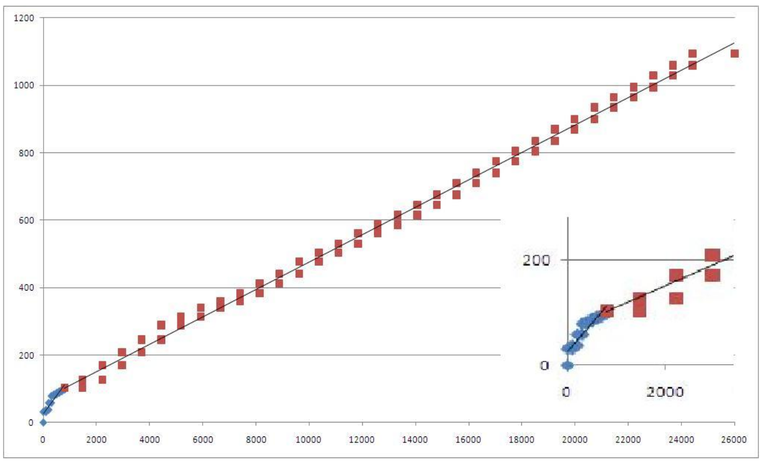


Figure 6: A piece-wise linear cost profile that is approximated by two linear functions

After depicting a general frame of the column generation model, it is believed that introducing the sets, the parameters, and the variables is beneficial for the reader to understand the model thoroughly.

### 4.2.1 Sets

Depending on the structure of the column generation model, the origins, the destinations, the hubs, the planning period, the arcs, and the set that contains the generated paths define the sets of the column generation model.

- The set List $= O \cup D \cup H$, define all the nodes in the system.
- The set $P = \{p_{i,k} \mid k \in K_i,\ (o_i,d_i) \in \text{OrgDes}\}$ contains all generated paths $p_{i,k}$, where $K_i$ is the index set of generated paths starting from origin $o_i \in O$ to its destination $d_i \in D$.

### 4.2.2 Parameters

The column generation method requires the weekly demands, the arc-path incidence matrix of the generated paths, the capacity of a truck in each link, the price of renting a full truck for each link, the dual values from the constraint groups (19) and (20) together with several data about the linear cost functions as input parameters.

Since the approximated cost functions consist of two different linear functions, the break-even point (denoted by $\text{Limit}_{a,b}$) where the model changes from the first linear cost function to the second is a parameter group to the model. Besides that, the slopes and the initial costs of the functions are the other parameters.

- $\text{Weight}_{i,l}$ is the amount of goods in kg's to be transported on week $w_l$ for all $i$ such that $(o_i,d_i) \in \text{OrgDes}$, and $l \in W$.
- $\text{ArcPath}_{a,b,k}$ is the arc-path incidence matrix for all paths $p_{i,k}$ such that $k \in K_i$, where $(o_i,d_i) \in \text{OrgDes}$ and $(a,b) \in A$. This matrix assigns 1 to the corresponding element if link $(a,b)$ is included in path $k$ and 0 otherwise.
- $\text{Limit}_{a,b}$ is the break-even point of cost function for the link $(a,b) \in A$. That is the value where the model starts using the second linear cost function for link $(a,b)$.
- $\text{Capacity}_{a,b}$ is the capacity of one truck on $(a,b) \in A$.
- $\text{Rate1}_{a,b}$ is the slope of the first linear function of link $(a,b) \in A$.
- $\text{Rate2}_{a,b}$ is the slope of the second linear function of link $(a,b) \in A$.
- $\text{Initial1}_{a,b}$ is the initial cost of the first linear function of link $(a,b) \in A$.
- $\text{Initial2}_{a,b}$ is the initial cost of the second linear function of link $(a,b) \in A$.
- $\text{Price}_{a,b}$ is the cost of renting one full truck on link $(a,b) \in A$.
- $\text{Dual1}_{a,b,l}$ is the dual variables retrieved from the constraint group (20) such that $(a,b) \in A$ and $l \in W$.
- $\text{Dual2}_{i,l}$ is the dual variables retrieved from the constraint group (19) such that $o_i \in O$ and $l \in W$.

### 4.2.3 Binary and Integer Variables

The column generation method is formulated as a relaxation of the main model in Section 4.1.6. Because of this relaxation, some of the rules and the restrictions introduced in Sections 2.1 and 2.2 are neglected, while some new rules are introduced.

Similar to the main model in Section 4.1.6, the variable that keeps track of the truck usage is introduced in the column generation phase. On the contrary, new variables that determine the active linear cost function and the active initial cost of the function are introduced for each link. Hence, the new binary and integer variables are as follows:

$$\text{Z1}_{a,b,l} = \begin{cases} 1, & \text{if the first linear cost function of the link } (a,b) \in A \\ & \text{during } l \in W \text{ is active in the } \textit{Master Problem}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{Z2}_{a,b,l} = \begin{cases} 1, & \text{if the second linear cost function of the link } (a,b) \in A \\ & \text{during } l \in W \text{ is active in the } \textit{Master Problem}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{Z1Sub}_{i,a,b,l} = \begin{cases} 1, & \text{if the first linear cost function of the link } (a,b) \in A \text{ during} \\ & l \in W \text{ is active in the } \textit{Sub-Problem } i \text{ such that } o_i \in O, \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{Z2Sub}_{i,a,b,l} = \begin{cases} 1, & \text{if the second linear cost function of the link } (a,b) \in A \text{ during} \\ & l \in W \text{ is active in the } \textit{Sub-Problem } i \text{ such that } o_i \in O, \\ 0, & \text{otherwise,} \end{cases}$$

- $\text{Truck}_{a,b,l}$ is the number of trucks utilized for link $(a,b) \in A$ during week $l \in W$ in the *Master Problem*.

$$\text{Truck}_{a,b,l} \geq 0, \quad \text{and integer}$$

- $\text{TruckSub}_{i,a,b,l}$ is the number of trucks utilized for link $(a,b) \in A$ during week $l \in W$ in the *Sub-Problem* $i$ such that $o_i \in O$.

$$\text{TruckSub}_{i,a,b,l} \geq 0, \quad \text{and integer}$$

### 4.2.4 Continuous Variables

Having introduced the binary and the integer variables, the column generation model still requires other variables. In the *Master Problem*, there is no restriction on the number of paths used to transport the goods from an origin to its destination. That is why, the *Master Problem* only requires the variable that keeps track of the amounts transported through a path. On the other hand, the *Sub-Problem* phase utilizes the links and tries to find paths that can be utilized to reduce the cost for the transportation between each origin-destination pair over the links. Hence, the *Sub-Problem* requires the variable that keeps track of the amounts transported through each link. Additionally, since the total costs of the *Master Problem* and the *Sub-Problem* are calculated depending on two linear functions for each link, the amounts that are transported using each cost function must be continuous variables.

- $\text{AMaster}_{i,k,l}$ is the amount transported in the *Master Problem* from the origin $o_i \in O$ through the path $k \in K_i$ on week $l \in W$ such that $(o_i,d_i) \in \text{OrgDes}$.

$$\text{AMaster}_{i,k,l} \geq 0$$

- $\text{ASub}_{i,a,b,l}$ is the amount transported in the *Sub Problem* $i$ through the link $(a,b) \in A$ on week $l \in W$ such that $o_i \in O$.

$$\text{ASub}_{i,a,b,l} \geq 0$$

- $\text{Piece1}_{a,b,l}$ is the amount transported in the *Master Problem* using the first linear cost function of link $(a,b) \in A$ on week $l \in W$.

$$\text{Piece1}_{a,b,l} \geq 0$$

- $\text{Piece2}_{a,b,l}$ is the amount transported in the *Master Problem* using the second linear cost function of link $(a,b) \in A$ on week $l \in W$.

$$\text{Piece2}_{a,b,l} \geq 0$$

- $\text{Piece1Sub}_{i,a,b,l}$ is the amount transported in the *Sub-Problem* $i$ using the first linear cost function of link $(a,b) \in A$ on week $l \in W$, such that $o_i \in O$.

$$\text{Piece1Sub}_{i,a,b,l} \geq 0$$

- $\text{Piece2Sub}_{i,a,b,l}$ is the amount transported in the *Sub-Problem* $i$ using the second linear cost function of link $(a,b) \in A$ on week $l \in W$ such that $o_i \in O$.

$$\text{Piece2Sub}_{i,a,b,l} \geq 0$$

### 4.2.5 The Master Problem

The *Master Problem* seeks the cheapest way to transport the demanded amount of goods using the paths generated so far, and calculates the dual values of the constraint groups (19) and (20). These dual values are then utilized to update the cost functions in the *Sub-Problem*. The *Master Problem* is initialized with a (basic) feasible solution and this set of paths is updated every time the *Master Problem* receives a new set of paths from the *Sub-Problem* phase.

The *Master Problem* consists of the following objective function and constraints.

The objective function is to minimize the total cost. The costs may occur either by transporting the goods using the first or the second linear cost function of each link or by utilizing another truck if the transported amount is above the capacity limit of a truck. When the first or second linear cost function is active, the respective initial costs are triggered. Hence, the objective function is to

$$
\begin{aligned}
\text{minimize} \quad & \sum_{l=1}^{L} \sum_{(a,b)\in A} \Big( \text{Piece1}_{a,b,l} \cdot \text{Rate1}_{a,b} + \text{Piece2}_{a,b,l} \cdot \text{Rate2}_{a,b} \\
& + \text{Z1}_{a,b,l} \cdot \text{Initial1}_{a,b} + \text{Z2}_{a,b,l} \cdot \text{Initial2}_{a,b} + \text{Truck}_{a,b,l} \cdot \text{Price}_{a,b} \Big), \quad l \in W
\end{aligned} \tag{18}
$$

The first constraint group of the *Master Problem* deals with the demands. The generated set of paths should be utilized to satisfy the demands. Additionally, this constraint group whose dual values are utilized to update the cost function in the *Sub-Problem*, is in accordance with the constraints (7c) in the restricted master problem (7) of MMCFP.

$$
\sum_{k\in K_i} \text{AMaster}_{i,k,l} = \text{Weight}_{i,l}, \quad o_i \in O, \quad l \in W \tag{19}
$$

The second constraint group of the *Master Problem* copes with the capacity of each link. Even though there is no limit on the capacities of links in practice, virtual capacities at each link are formed by this constraint group. It is assumed that the capacity of a link is equal to the amount conveyed through that link on each week. Therefore, this constraint group becomes similar to the constraints (7b) in the restricted master problem (7) of MMCFP.

$$
\begin{aligned}
\sum_{i:o_i\in O} \quad \sum_{k\in K_i} \text{ArcPath}_{a,b,k} \cdot \text{AMaster}_{i,k,l} = {} & \text{Truck}_{a,b,l} \cdot \text{Capacity}_{a,b} \\
& + \text{Piece1}_{a,b,l} + \text{Piece2}_{a,b,l}, \quad (a,b) \in A, \quad l \in W,
\end{aligned} \tag{20}
$$

The last two constraint groups of this part of the model are used to determine whether the first or the second cost function of each link is active on week $l$.

$$
\text{Piece1}_{a,b,l} \le \text{Limit}_{a,b} \cdot \text{Z1}_{a,b,l}, \quad (a,b) \in A, \quad l \in W \tag{21}
$$

$$
\text{Piece2}_{a,b,l} \le \text{Capacity}_{a,b} \cdot \text{Z2}_{a,b,l}, \quad (a,b) \in A, \quad l \in W \tag{22}
$$

#### 4.2.6 The Sub-Problem

The *Sub-Problem* runs over each origin-destination pair and generates columns (i.e., paths) using the links in the model. It initializes, as it receives the dual values from the *Master Problem*, and recalculates the costs in the objective function using these values. The *Sub-Problem* includes an objective function and four constraint groups for each origin.

The objective function is very similar to the one in the *Master Problem*, (18), except that the dual values affect the total cost and it is calculated over only one origin. The objective function for each $i$ such that $o_i \in O$ is to

$$\begin{aligned} \text{minimize} \quad \sum_{l=1}^{L} \sum_{(a,b)\in A} \Big( & \text{Piece1Sub}_{i,a,b,l} \cdot \text{Rate1}_{a,b} \text{Piece2Sub}_{i,a,b,l} \cdot \text{Rate2}_{a,b} \\ & + \text{Z1Sub}_{i,a,b,l} \cdot \text{Initial1}_{a,b} + \text{Z2Sub}_{a,b,l} \cdot \text{Initial2}_{a,b} \\ & + \text{TruckSub}_{i,a,b,l} \cdot \text{Price}_{a,b} + \text{ASub}_{i,a,b,l} \cdot \text{Dual1}_{a,b,l} - \text{Dual2}_{i,l} \Big) \end{aligned} \tag{23}$$

The first constraint group makes sure that the flow through the nodes is balanced, i.e., the flow defines (a number of) paths for each origin-destination pair.

$$\sum_{b\in \text{List}:(a,b)\in A} \text{ASub}_{i,a,b,l} - \sum_{b\in \text{List}:(b,a)\in A} \text{ASub}_{i,b,a,l} = \begin{cases} \text{Weight}_{i,l}, & \text{if } a \in O, \\ -\text{Weight}_{i,l}, & \text{if } a \in D, \\ 0, & \text{if } a \in \text{List}\backslash(O \cup D). \end{cases} \tag{24}$$

If the constraints (24) are applied to the network in Figure 5, assuming that one unit is to be transported, then the constraints (24) appear as

$$\begin{aligned} x_{s1} + x_{s3} &= 1, \\ x_{12} - x_{s1} - x_{31} &= 0, \\ x_{24} + x_{2t} - x_{12} &= 0, \\ x_{31} + x_{34} - x_{s3} &= 0, \\ x_{4t} - x_{24} - x_{34} &= 0, \\ -x_{2t} - x_{4t} &= -1. \end{aligned}$$

The second constraint group is very similar to the capacity constraint group, (20), in the *Master Problem.* The virtual capacities of the links are calculated by the amount of trucks utilized and the amounts transported using the first or the second approximation function of the links. For each $i$ such that $o_i \in O$, the constraints read

$$\begin{aligned} \text{ASub}_{i,a,b,l} = & \text{TruckSub}_{i,a,b,l} \cdot \text{Capacity}_{a,b} + \text{Piece1Sub}_{i,a,b,l} \\ & + \text{Piece2Sub}_{i,a,b,l}, \quad (a,b) \in A, \quad l \in W. \end{aligned} \tag{25}$$

The last two constraint groups of this part of the model are used to determine whether the first or the second cost function of each link is active on week $l$; they read

$$\text{Piece1Sub}_{i,a,b,l} \leq \text{Limit}_{a,b} \cdot \text{Z1}_{a,b,l}, \quad o_i \in O, \quad (a,b) \in A, \quad l \in W, \tag{26}$$

$$\text{Piece2Sub}_{i,a,b,l} \leq \text{Capacity}_{a,b} \cdot \text{Z2}_{a,b,l}, \quad o_i \in O, \quad (a,b) \in A, \quad l \in W. \tag{27}$$

### 4.2.7 The Complete Column Generation Model

The column generation model is summarized below.
**Master Problem**

$$\text{minimize} \quad \sum_{l=1}^{L} \sum_{(a,b)\in A} \Big( \text{Piece1}_{a,b,l} \cdot \text{Rate1}_{a,b} + \text{Piece2}_{a,b,l} \cdot \text{Rate2}_{a,b}$$

$$+ \text{Z1}_{a,b,l} \cdot \text{Initial1}_{a,b} + \text{Z2}_{a,b,l} \cdot \text{Initial2}_{a,b} + \text{Truck}_{a,b,l} \cdot \text{Price}_{a,b} \Big),$$

$$\text{subject to} \quad \sum_{k\in K_i} \text{AMaster}_{i,k,l} = \text{Weight}_{i,l}, \quad o_i \in O, \quad l \in W,$$

$$\sum_{i:o_i\in O} \sum_{k\in K_i} \text{ArcPath}_{a,b,k} \cdot \text{AMaster}_{i,k,l} = \text{Truck}_{a,b,l} \cdot \text{Capacity}_{a,b} + \text{Piece1}_{a,b,l} + \text{Piece2}_{a,b,l},$$

$$(a,b) \in A, \quad o_i \in O, \quad l \in W,$$

$$\text{Piece1}_{a,b,l} \leq \text{Limit}_{a,b} \cdot \text{Z1}_{a,b,l}, \quad (a,b) \in A, \quad l \in W,$$

$$\text{Piece2}_{a,b,l} \leq \text{Capacity}_{a,b} \cdot \text{Z2}_{a,b,l}, \quad (a,b) \in A, \quad l \in W,$$

$$\text{Piece1}_{a,b,l} \geq 0, \quad (a,b) \in A, \quad l \in W,$$

$$\text{Piece2}_{a,b,l} \geq 0, \quad (a,b) \in A, \quad l \in W,$$

$$\text{Z1}_{a,b,l} \in \{0,1\}, \quad (a,b) \in A, \quad l \in W,$$

$$\text{Z2}_{a,b,l} \in \{0,1\}, \quad (a,b) \in A, \quad l \in W,$$

$$\text{Truck}_{a,b,l} \geq 0, \quad \text{and integer}, \quad (a,b) \in A, \quad l \in W,$$

$$\text{AMaster}_{i,k,l} \geq 0, \quad k \in K_i, \quad o_i \in O, \quad l \in W.$$

**Sub-Problem**

For each $o_i \in O$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{l=1}^{L} \sum_{(a,b)\in A} \Big(\text{Piece1Sub}_{i,a,b,l} \cdot \text{Rate1}_{a,b} + \text{Piece2Sub}_{i,a,b,l} \cdot \text{Rate2}_{a,b} \\
& \qquad + \text{Z1Sub}_{i,a,b,l} \cdot \text{Initial1}_{a,b} + \text{Z2Sub}_{a,b,l} \cdot \text{Initial2}_{a,b} \\
& \qquad + \text{TruckSub}_{i,a,b,l} \cdot \text{Price}_{a,b} + \text{ASub}_{i,a,b,l} \cdot \text{Dual1}_{a,b,l} - \text{Dual2}_{i,l}\Big), \\
\text{subject to} \quad & \text{ASub}_{i,a,b,l} = \text{TruckSub}_{i,a,b,l} \cdot \text{Capacity}_{a,b} + \text{Piece1Sub}_{i,a,b,l} + \text{Piece2Sub}_{i,a,b,l}, \\
& \qquad (a,b) \in A, \quad l \in W,
\end{aligned}
$$

$$
\begin{aligned}
\text{Piece1Sub}_{i,a,b,l} &\leq \text{Limit}_{a,b} \cdot \text{Z1}_{a,b,l}, \quad (a,b) \in A, \quad l \in W, \\
\text{Piece2Sub}_{i,a,b,l} &\leq \text{Capacity}_{a,b} \cdot \text{Z2}_{a,b,l}, \quad (a,b) \in A, \quad l \in W, \\
\text{Piece1Sub}_{i,a,b,l} &\geq 0, \quad (a,b) \in A, \quad l \in W, \\
\text{Piece2Sub}_{i,a,b,l} &\geq 0, \quad (a,b) \in A, \quad l \in W, \\
\text{Z1Sub}_{i,a,b,l} &\in \{0,1\}, \quad (a,b) \in A, \quad l \in W, \\
\text{Z2Sub}_{a,b,l} &\in \{0,1\}, \quad (a,b) \in A, \quad l \in W, \\
\text{TruckSub}_{i,a,b,l} &\geq 0, \quad \text{and integer}, \quad (a,b) \in A, \quad l \in W, \\
\text{ASub}_{i,a,b,l} &\geq 0, \quad k \in K_i, \quad (a,b) \in A, \quad l \in W,
\end{aligned}
$$

$$
\sum_{b\in \text{List}:(a,b)\in A} \text{ASub}_{i,a,b,l} - \sum_{b\in \text{List}:(b,a)\in A} \text{ASub}_{i,b,a,l} = \begin{cases} \text{Weight}_{i,l}, & \text{if } a \in O, \\ -\text{Weight}_{i,l}, & \text{if } a \in D, \\ 0, & \text{if } a \in \text{List}\backslash(O \cup D). \end{cases}
$$

# 5 Tests and Results

This section consists of a presentation of the test cases, the tests performed, results retrieved from these, and analysis of the results. The model introduced in Section 4.1 involves several different dimensions and a large amount of variables. There is approximately one million binary variables required just to model the pieces of the cost functions of all the links. Similarly, if one wants to solve the problem over all possible paths, then approximately seven million binary variables are required to model all possible paths. Moreover, the number of variables grow multiplicatively depending on the planning period. Therefore, it is believed that establishing different test cases by varying the planning period and the set of possible paths may help to explain the results and express the effect of the number of variables thoroughly.

## 5.1 The Test Cases, Results and Analysis

This section consists of five subsections. Each subsection begins with introducing a new group of tests which is followed by their results and analysis. All the results presented in this section, apart from the results of the column generation model, have been retrieved from CPLEX just before computer runs out of memory. Hence, the results may not present the optimal solutions, but relatively good solutions together with upper and lower bounds on the optimal solutions.

### 5.1.1 The Column Generation Model

The column generation model which is presented in Section 4.2.7 is solved and analyzed in this section. The restricted master problem which is called the *Master Problem* in Section 4.2.7 is initialized by utilizing the direct paths and the paths that include one hub. Afterwards, the column generation model has been run five times. Table 1 shows how many paths have been generated at each iteration.

| Iteration no | Number of paths generated | Optimal value of restricted master problem |
|---|---|---|
| 0 | 171 | 122113 |
| 1 | 48 | 122014 |
| 2 | 23 | 121815 |
| 3 | 29 | 121760 |
| 4 | 37 | 121612 |
| 5 | 26 | 121470 |

Table 1: Number of generated paths at each iteration of the column generation algorithm and the corresponding objective values.

As discussed in Section 4.2, the generated paths are added to the set of paths that have already been generated, and the *Master Problem* runs over an extended set of paths. Therefore, the total cost calculated in the *Master Problem* keeps decreasing as the number of generated paths increase. Figure 7 depicts this change in the total cost.

Table 1 shows that the column generation model has been iterated five times and total of 163 paths have been generated. That is, a total of 334 paths have been

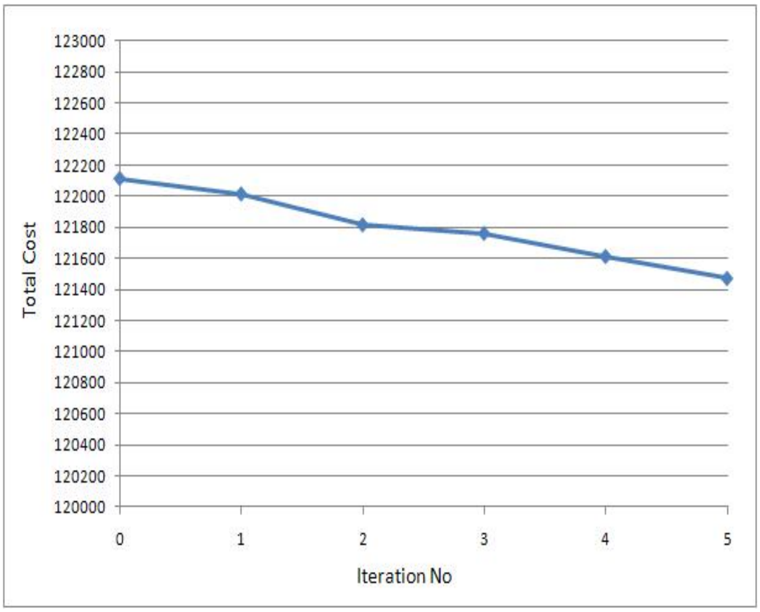


Figure 7: Total cost of the optimal solution of the *Master Problem* at each iteration

generated, together with the initial paths which has been provided to the *Master Problem* for initialization. This is approximately 0.87% of all possible paths. One could assert that the column generation method should have been iterated many more times and that many paths should have been generated.

On the contrary, Figure 7 shows that the decrease in the total cost of *Master Problem* is only 0.5% after generating total of 334 paths. In general, the decrease in the objective value is usually large at early stages of the column generation models. After a number of iterations, the objective function value typically does not decrease much [7]. Therefore, Figure 7 puts forward that the initial paths that have been provided to the *Master Problem* includes a solution that is very close to the optimum, since the slope of the function is not steep. Hence, iterating the column generation model more and generating new columns would probably not yield much improvement.

Moreover, the model presented in Section 4.1.6 becomes much harder to solve, as the number of the paths increases. Therefore, as the amount of decrease in the objective value indicates, it is possible to utilize these 334 paths to be able to find a very good solution to the model in Section 4.1.6.

### 5.1.2 The Cost Function Approximations

Each cost function consists of many small pieces, varying from 60 to 83 pieces, so it is believed that these functions could be further investigated by approximating them from above and from below. In doing this approximation, the number of pieces in each cost function is decreased by 50%. Figures 8 and 9 describe how the approximations from above and below are made.

This test group involves two different cases, one for the cost functions that approximate the actual cost functions from above, and one for the cost functions that approximate the actual cost functions from below. These two tests have been

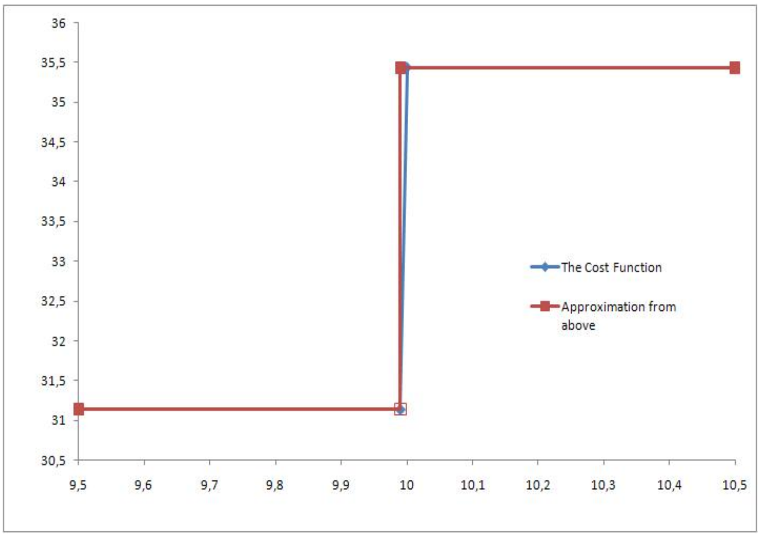


Figure 8: An example of the approximation from above of two pieces of a cost function

performed by utilizing the paths generated after five iterations in column generation algorithm. Table 2 presents the total costs of each case together with the respective lower bounds and optimality gaps.

| Approximation | Best solution found (SEK) | Lower bound on the optimal value (SEK) | Optimality gap |
|---|---|---|---|
| From above | 130754 | 111586 | 14.7 % |
| From below | 123868 | 109873 | 11.3 % |

Table 2: Results retrieved from the main model for different approximations of the actual cost functions

These tests have been performed to explain the differences between the approximated cost functions. The results show that the total cost of the best solution found decreases by 7000SEK as the approximations from above are replaced by the approximations from below. Since one function group approximates the cost from above, while the other group approximates them below, the results show that the optimal value lies in the interval [109873,130754].

One other interesting result is that when the cost functions are approximated from below, the optimality gap decreases from 14.66% to 11.3%. Even though, this result needs further investigation by utilizing different problem instances, one may assert that the approximations from below should be utilized, if the optimality is the ultimate goal.

### 5.1.3 Different Planning Periods

This test group focused on varying the planning period. The planning periods have been defined to be 3,7,13,20 and 26 weeks, respectively, and these cases have been solved by utilizing the paths that have been generated by the column generation method in Section 5.1.1 and the cost functions that approximate the actual cost

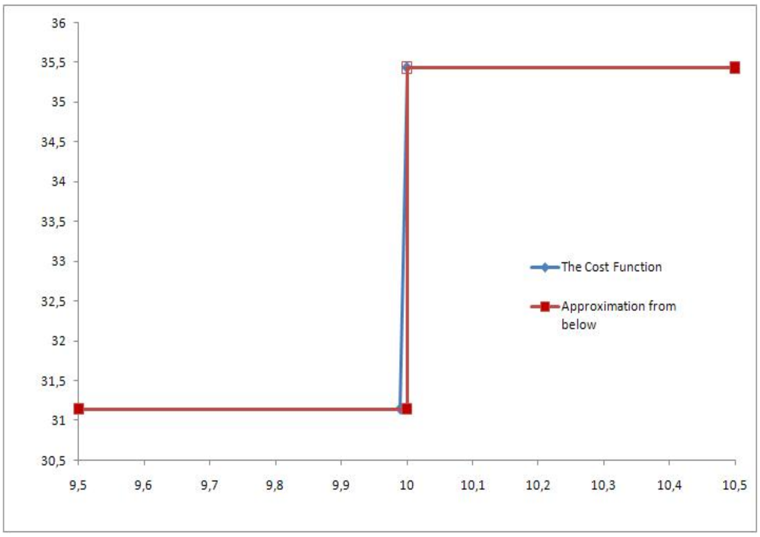


Figure 9: An example of the approximation from below of two pieces of a cost function

functions from above (desrcibed in Section 5.1.2). Table 3 presents the total costs for the best solutions of the different planning periods together with the respective lower bounds and the optimality gaps.

| Planning period | Best solution found (SEK) | Lower bound on the optimal value (SEK) | Optimality gap | Total cost per week (SEK) |
|---|---|---|---|---|
| 3 weeks | 20232 | 18591 | 8.1 % | 6744 |
| 7 weeks | 47768 | 43259 | 9.2 % | 6824 |
| 13 weeks | 82838 | 73473 | 11.3 % | 6372 |
| 20 weeks | 103468 | 91248 | 12.0 % | 5173 |
| 26 weeks | 130754 | 111586 | 14.7 % | 5029 |

Table 3: Results retrieved from the model presented in Section 4.1.6 for different planning periods

Each origin-destination pair has only one active path. Figures 10 and 11 show the solutions to 3 weeks of planning on a table and on a map, respectively. For the solutions of the other planning periods, see Appendix A.

The tests which include different planning periods have been performed to describe the changes in the solution set. Additionally, these tests have been performed in order to investigate whether a shorter and a longer planning period yield similar results or not.

First of all, Table 3 presents that the optimality gap increases as the time-span increases. This is a reasonable result, since the number of variables increase multiplicatively, as the number of weeks increase, which makes the problem much harder to solve.

In addition to that, if the solutions of the 20 weeks case are compared to those of the 26 weeks case, there appears to be some similarities, especially in selecting the hubs. Actually, 21 out of 31 origin points use the same paths for these two different cases. On the other hand, none of these paths are utilized on the same

| Origins | Via | Days to Send | Fraction to Send |
|---|---|---|---|
| Amotfors | Goteborg | Friday | 1 |
| Anderstorp | - | Thursday | 1 |
| Angelholm | - | Sunday | 1 |
| Arboga | Vargarda | Monday | 1 |
| Atvidaberg | Vargarda | Monday | 1 |
| Bredaryd | - | Tuesday | 1 |
| Dalsjofors | Vargarda | Monday | 1 |
| Dobeln | - | Thursday | 1 |
| Eket | - | Sunday | 1 |
| Falkenberg | - | Monday | 1 |
| Forsheda | - | Friday | 1 |
| Goteborg | - | Saturday | 1 |
| Hallstahammar | Varnamo | Tuesday | 1 |
| Halmstad | Goteborg | Friday | 1 |
| Hassleholm | - | Thursday | 1 |
| Herrljunga | - | Wednesday | 1 |
| Huskvarna | Vargarda | Monday | 1 |
| Kinnared | - | Monday | 1 |
| Klippan | Malmo | Wednesday | 1 |
| Korntal | - | Tuesday | 1 |
| Malmo | - | Thursday | 1 |
| Molndal | Goteborg | Friday | 1 |
| Norrkoping | Vargarda | Monday | 1 |
| Orkelljunga | - | Monday | 1 |
| Smalandsstenar | - | Tuesday | 1 |
| Torekov | Varnamo | Tuesday | 1 |
| Torsas | Varnamo | Tuesday | 1 |
| Ulricehamn | Goteborg | Friday | 1 |
| Vargarda | - | Tuesday | 1 |
| Varnamo | - | Wednesday | 1 |
| Vimmerby | - | Tuesday | 1 |

Figure 10: Solution to the planning over three weeks. The destination for all the origins is Helsinki.

day for the two cases. Moreover, if the solutions of the 7 weeks case are compared to the 26 weeks case, 20 out of 31 origin points use the same paths, while none of the departure days overlap for these two different cases. These results point out that the results from a smaller time-span may, to some extent, describe the results of a larger time-span. Therefore, if a case is very large, then the results from a subset of the weeks could be utilized and these results could still yield a relatively good solution. Similarly, results retrieved from a smaller time-span of a case could as well be utilized as an initial basis to the main optimization model introduced in Section 4.1.6 so that the main problem would start from a relatively good solution and thus may find a very good solution in a shorter computational time.

One other interesting result is that the number of departure days per week and origin-destination pair tend to one, as the total cost gets closer to the optimal value. In order to further investigate this finding, the main model has been run one more time with all the generated paths, but letting the number of departure days during each week be at most one. In doing that, the total cost found was 134532SEK, which is approximately 2000SEK more than the total cost found by utilizing all the generated paths and the above approximations of the cost functions (see Table 4 in Section 5.1.4). This shows that utilizing a smaller set of departure days could still provide a relatively good result. This finding is actually very useful, if the problem is very large to solve.

### 5.1.4 Changing the Set of Paths

This test group focused on varying the sets of possible paths and three different cases have been defined. The first case has run over the direct paths, while the second case have used the direct paths and the paths that include one hub. Besides these, the last case has utilized all the paths generated by the column generation model

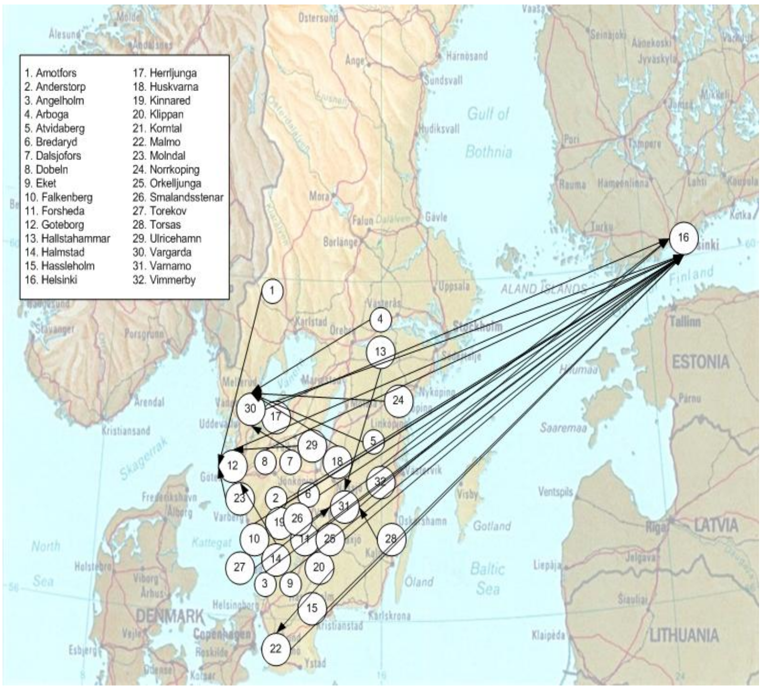


Figure 11: A geographical picture of the paths that define the best solution found to three weeks of planning on the map

(see Table 1). All of these cases have been solved by utilizing the cost functions that approximate the actual cost functions from above. Table 4 presents the total costs of the different sets together with the lower bounds and the optimality gaps.

| Elements of the path set | Best solution found (SEK) | Lower bound on the optimal value (SEK) | Optimality gap |
|---|---|---|---|
| Direct relations | 146042 | 125286 | 14.2 % |
| Direct relations + relations with 1 hub | 132883 | 108938 | 18.0 % |
| The paths generated by the column generation method | 130754 | 111586 | 14.7 % |

Table 4: Results retrieved from the main model for different sets of possible paths

The tests, which have been performed by changing the set of possible paths, are aimed to describe the changes in the total cost. Table 4 depicts the results to these cases. The most significant information that is retrieved from this table is that as the amount of paths increase, the total cost of the main problem decreases. This shows that the generated paths are of good quality with respect to the objective function.

Besides that, the results underline the importance of the hubs, since the total cost is reduced by approximately 9% when the relations with one hub are utilized. In accordance with this reduction, the total cost is reduced by another 1.6% when all the paths that have been generated by the column generation method are utilized. This shows that enlarging the solution set by increasing the number of possible paths return better results. On the other hand, one should still be very careful about this proposition, since it may become very hard to even compute a solution if too many paths are utilized at the same time. Hence, analyzing the total cost retrieved from the *Master Problem* after few iterations may be an effective way in order to balance the size of the solution set and the size of the problem. This actually is reasonable, because the behavior of the total cost retrieved from the *Master Problem* somewhat explains the possible gain in the main model introduced in Section 4.1.6.

### 5.1.5 The Paths Generated by Proxio

In this test, the paths computed by Proxio that yield the lowest total cost, are utilized as the set of possible paths in the main model introduced in Section 4.1.6 together with the actual cost functions introduced in Section 2.2. Figure 12 shows this set of paths, and Table 5 shows the results retrieved from the optimization model by utilizing these paths.

| Origins | Via | Destination |
|---|---|---|
| Amotfors | Goteborg | Helsinki |
| Anderstorp | Vargarda | Helsinki |
| Angelholm | Malmo | Helsinki |
| Arboga | Vargarda | Helsinki |
| Atvidaberg | Vargarda | Helsinki |
| Bredaryd | - | Helsinki |
| Dalsjofors | Vargarda | Helsinki |
| Dobeln | - | Helsinki |
| Eket | Vargarda | Helsinki |
| Falkenberg | - | Helsinki |
| Forsheda | Vargarda | Helsinki |
| Goteborg | - | Helsinki |
| Hallstahammar | Vargarda | Helsinki |
| Halmstad | Vargarda | Helsinki |
| Hassleholm | Vargarda | Helsinki |
| Herrljunga | Vargarda | Helsinki |
| Huskvarna | Vargarda | Helsinki |
| Kinnared | - | Helsinki |
| Klippan | Malmo | Helsinki |
| Korntal | - | Helsinki |
| Malmo | - | Helsinki |
| Molndal | Vargarda | Helsinki |
| Norrkoping | Vargarda | Helsinki |
| Orkelljunga | Goteborg | Helsinki |
| Smalandsstenar | Goteborg | Helsinki |
| Torekov | Malmo | Helsinki |
| Torsas | Vargarda | Helsinki |
| Ulricehamn | Vargarda | Helsinki |
| Vargarda | - | Helsinki |
| Varnamo | Vargarda | Helsinki |
| Vimmerby | Vargarda | Helsinki |

Figure 12: The paths (one path for each origin-destination pair) that have been retrieved from Proxio

| Best solution found (SEK) | Lower bound on the optimal value (SEK) | Optimality gap |
|---|---|---|
| 141121 | 119849 | 17.8 % |

Table 5: Results retrieved from the optimization model by utilizing the paths retrieved from Proxio

The company mentioned that the total cost for the 26 weeks case computed by the Proxio Optimizer is 120200SEK. This objective value is approximately 17.4%

(see Table 5) less than the total cost of the best solution found by the optimization model before CPLEX runs out of memory. This shows that the optimization model has not been able to produce a good upper bound by utilizing the paths retrieved from Proxio. On the other hand, the lower bound calculated by the optimization model shows that Proxio Optimizer calculates a feasible solution that is at most 0.3% (see Table 5) more expensive than the cheapest possible solution, utilizing only the set of paths in Figure 12.

Moreover, the total cost calculated by Proxio is approximately 8.07% (see Table 2) less than the total cost which has been calculated by utilizing the cost functions that approximate the actual cost functions from above. Additionally, the total cost calculated by Proxio is approximately 3% (see Table 2) less than the total cost which has been calculated by utilizing the cost functions that approximate the actual cost functions from below.

On the other hand, one should as well keep in mind that the optimization model has not been able to solve the problem to optimality, and the lower bounds for the approximations from above and below are 111586.2 and 109873.6, respectively. Hence, there still is a possibility to improve the results of the optimization model.

# 6 Discussion and Future Work

This master thesis has been performed to evaluate the efficiency of the Proxio Optimizer by developing and testing a mathematical optimization model. The results show that the company has developed a very effective optimization engine. The Proxio Optimizer produces better feasible solutions than the optimization model.

Even though the Proxio Optimizer produces better results, the solution from the optimization model may still be improved, since there is an optimality gap larger than 10% for the cases investigated. Hence, any attempt to improve the optimization algorithm should actually focus on reducing this gap. Besides that, reducing the gap may help the company much to find out the capabilities of their software.

Moreover, having developed a column generation algorithm is still regarded as an important achievement, since the column generation algorithm may help the company to deal with much larger cases easily and efficiently. On the contrary, one may still work on the column generation model for further improvements, since this algorithm utilizes approximations of the cost functions which may limit the accuracy of the column generation phase. E.g., one may incorporate the column generation scheme in a branch-and-bound algorithm for integer programs, a so called branch-and-price algorithm, or one could employ so-called column dropping in order to reduce the size of the master problem.

In addition to that, the company has several rules such as defining only one path for each origin destination point. This rule may actually be much more flexible so that customers can select from a group of paths. On the other hand, this may have an adverse affect on the total cost, since the combination of several paths may cause to the total cost to rise.

# References

[1] Jonnson, P. 2008. Logistic and supply chain management. McGraw-Hill Education.

[2] Proxio Inc. *Proxio Optimizer*. 03 May 2010, http://www.proxio.se/eng.

[3] Stefansson, G., 2009. *Introduction to logistics and supply chain management.* Retrieved from CTH database.

[4] Dov, D., Tzvi R., and Shenhar, A.J., February 2003. An empirical analysis of the relationship between project planning and project success. *International Journal of Project Management*, *21*(2), p.89-95.

[5] Matoušek, J., and Gärtner, B. (2007). *Understanding and using linear programming.* Springer Berlin Heidelberg.

[6] Wolsey, L.A. (1998). *Integer programming.* Wiley-Interscience.

[7] Desaulniers, G., Desrosiers J., and Solomon, M.M. (2005). *Column generation.* Springer US.

[8] Bazaraa, M.S., Jarvis, J.J., and Sherali, H.D. (1990). *Linear programming and network flows.* John Wiley & Sons.

[9] Lawler, E.L. (2001). *Combinatorial optimization: network and matroids.* Dover Publications.

[10] Fourer, R., Gay, D.M., and Kernighan B.W. (2003). *AMPL A modeling language for mathematical programming.* Brooks/Cole.

[11] IBM ILOG CPLEX V12.1. (2009). *Users Manual.* ILOG. 01 April 2010, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[12] Nash, S.G., and Sofer, A. (1996). *Linear and Nonlinear Programming* McGraw-Hill.

# Appendices

## Appendix A Solutions for Different Planning Periods

| Origins | Via | Days to Send | Fraction to Send |
|---|---|---|---|
| Amotfors | Goteborg | Tuesday | 1 |
| Anderstorp | Varnamo | Friday | 1 |
| Angelholm | - | Monday | 1 |
| Arboga | Vargarda | Friday | 1 |
| Atvidaberg | Vargarda | Saturday | 1 |
| Bredaryd | - | Wednesday | 1 |
| Dalsjofors | Vargarda | Thursday | 1 |
| Dobeln | - | Monday | 1 |
| Eket | Varnamo | Friday | 1 |
| Falkenberg | - | Monday | 1 |
| Forsheda | Varnamo | Friday | 1 |
| Goteborg | - | Wednesday | 1 |
| Hallstahammar | Vargarda | Thursday | 1 |
| Halmstad | Goteborg | Tuesday | 1 |
| Hassleholm | Goteborg | Tuesday | 1 |
| Herrljunga | - | Thursday | 1 |
| Huskvarna | Vargarda | Saturday | 1 |
| Kinnared | - | Wednesday | 1 |
| Klippan | Goteborg | Tuesday | 1 |
| Korntal | - | Monday | 1 |
| Malmo | - | Sunday | 1 |
| Molndal | Goteborg | Tuesday | 1 |
| Norrkoping | Vargarda | Friday | 1 |
| Orkelljunga | Goteborg | Tuesday | 1 |
| Smalandsstenar | Varnamo | Friday | 1 |
| Torekov | Varnamo | Friday | 1 |
| Torsas | Varnamo | Friday | 1 |
| Ulricehamn | Vargarda | Saturday | 1 |
| Vargarda | - | Fri - Sat - Sun | 0.36 - 0.34 - 0.30 |
| Varnamo | - | Saturday | 1 |
| Vimmerby | Varnamo | Friday | 1 |

Figure 13: Solution to the planning over seven weeks. The destination for all the origins is Helsinki.

| Origins | Via | Days to Send | Fraction to Send |
|---|---|---|---|
| Amotfors | Vargarda | Friday | 1 |
| Anderstorp | Varnamo | Monday | 1 |
| Angelholm | - | Saturday | 1 |
| Arboga | Vargarda | Tuesday | 1 |
| Atvidaberg | Vargarda | Tuesday | 1 |
| Bredaryd | - | Thursday | 1 |
| Dalsjofors | Vargarda | Tuesday | 1 |
| Dobeln | - | Monday | 1 |
| Eket | Varnamo | Monday | 1 |
| Falkenberg | - | Monday | 1 |
| Forsheda | Varnamo | Monday | 1 |
| Goteborg | - | Wednesday | 1 |
| Hallstahammar | - | Friday | 1 |
| Halmstad | Goteborg | Tuesday | 1 |
| Hassleholm | Goteborg | Tuesday | 1 |
| Herrljunga | - | Thursday | 1 |
| Huskvarna | Vargarda | Tuesday | 1 |
| Kinnared | - | Saturday | 1 |
| Klippan | Goteborg | Tuesday | 1 |
| Korntal | - | Wednesday | 1 |
| Malmo | - | Saturday | 1 |
| Molndal | Goteborg | Tuesday | 1 |
| Norrkoping | Goteborg | Tuesday | 1 |
| Orkelljunga | Goteborg | Tuesday | 1 |
| Smalandsstenar | Varnamo | Monday | 1 |
| Torekov | Varnamo | Monday | 1 |
| Torsas | - | Saturday | 1 |
| Ulricehamn | Vargarda | Tuesday | 1 |
| Vargarda | - | Mon - Wed | 0.3 - 0.7 |
| Varnamo | - | Tuesday | 1 |
| Vimmerby | Varnamo | Monday | 1 |

Figure 14: Solution to the planning over thirteen weeks. The destination for all the origins is Helsinki.

| Origins | Via | Days to Send | Fraction to Send |
|---|---|---|---|
| Amotfors | Vargarda | Tuesday | 1 |
| Anderstorp | Varnamo | Monday | 1 |
| Angelholm | - | Saturday | 1 |
| Arboga | Vargarda | Tuesday | 1 |
| Atvidaberg | Vargarda | Wednesday | 1 |
| Bredaryd | - | Wednesday | 1 |
| Dalsjofors | Vargarda | Wednesday | 1 |
| Dobeln | - | Monday | 1 |
| Eket | Varnamo | Monday | 1 |
| Falkenberg | Goteborg | Saturday | 1 |
| Forsheda | Varnamo | Monday | 1 |
| Goteborg | - | Saturday | 1 |
| Hallstahammar | - | Wednesday | 1 |
| Halmstad | Goteborg | Saturday | 1 |
| Hassleholm | Goteborg | Saturday | 1 |
| Herrljunga | - | Monday | 1 |
| Huskvarna | Vargarda | Wednesday | 1 |
| Kinnared | - | Tuesday | 1 |
| Klippan | Goteborg | Saturday | 1 |
| Korntal | - | Friday | 1 |
| Malmo | - | Thursday | 1 |
| Molndal | Goteborg | Saturday | 1 |
| Norrkoping | Vargarda | Tue - Wed | 0.4 - 0.6 |
| Orkelljunga | Goteborg | Saturday | 1 |
| Smalandsstenar | Varnamo | Mon - Wed | 0.56 - 0.44 |
| Torekov | Varnamo | Monday | 1 |
| Torsas | - | Monday | 1 |
| Ulricehamn | Vargarda | Tuesday | 1 |
| Vargarda | - | Wed - Thu | 0.65 - 0.35 |
| Varnamo | - | Tuesday | 1 |
| Vimmerby | Varnamo | Monday | 1 |

Figure 15: Solution to the planning over twenty weeks. The destination for all the origins is Helsinki.

| Origins | Via | Days to Send | Fraction to Send |
|---|---|---|---|
| Amotfors | Vargarda | Friday | 1 |
| Anderstorp | - | Sunday | 1 |
| Angelholm | Halmstad | Saturday | 1 |
| Arboga | Vargarda | Thursday | 1 |
| Atvidaberg | Varnamo | Sunday | 1 |
| Bredaryd | - | Tue - Fri | 0.35 - 0.65 |
| Dalsjofors | Vargarda | Friday | 1 |
| Dobeln | - | Sunday | 1 |
| Eket | Halmstad | Saturday | 1 |
| Falkenberg | Halmstad | Saturday | 1 |
| Forsheda | Varnamo | Sunday | 1 |
| Goteborg | - | Tuesday | 1 |
| Hallstahammar | Vargarda | Monday | 1 |
| Halmstad | - | Sunday | 1 |
| Hassleholm | - | Saturday | 1 |
| Herrljunga | - | Wednesday | 1 |
| Huskvarna | Vargarda | Thursday | 1 |
| Kinnared | - | Tue - Thu | 0.88 - 0.12 |
| Klippan | Hassleholm | Friday | 1 |
| Korntal | - | Saturday | 1 |
| Malmo | - | Saturday | 1 |
| Molndal | Goteborg | Monday | 1 |
| Norrkoping | Vargarda | Friday | 1 |
| Orkelljunga | Halmstad | Saturday | 1 |
| Smalandsstenar | Varnamo | Sunday | 1 |
| Torekov | Halmstad | Saturday | 1 |
| Torsas | Varnamo | Sunday | 1 |
| Ulricehamn | Vargarda | Friday | 1 |
| Vargarda | - | Tue - Fri - Sat | 0.31 - 0.38 - 0.31 |
| Varnamo | - | Monday | 1 |
| Vimmerby | Varnamo | Sunday | 1 |

Figure 16: Solution to the planning over twenty six weeks. The destination for all the origins is Helsinki.