Aalto University

MS-E2177 SEMINAR ON CASE STUDIES IN OPERATIONS RESEARCH: FINAL REPORT

Posti Group Oyj: Integrated cost optimization of Parcel and eCommerce supply chain

Veli Kangas
Jussi Leppinen
Juuso Leskinen
Tuomas Suominen (PM)

June 6, 2020

Contents

Chapter 1 Introduction

1.1 Background

Logistics is concerned with the organization, movement and storage of material and people. The term logistics was first used by the military to describe the activities associated with maintaining a fighting force in the field and, in its narrowest sense, describes the housing of troops. Over the years the meaning of the term has gradually generalized to cover business and service activities [11].

Business requirements have however evolved beyond logistics. Logistics may be regarded as an integral part of a wider supply chain management concept. Supply chain management attempts to integrate all levels of an end-to-end process from the procurement of raw materials to manufacturing to logistics and product delivery into a single whole. A systemic view of the supply chain provides a means for profit and revenue optimization for businesses through system modeling and mathematical programming approaches [12].

In this report we discuss a mathematical optimization approach for the integrated planning of parcel and eCommerce business in the supply chain owned and operated by Posti Group.

1.2 Motivation

Posti Group is the leading postal and logistics service company in Finland, in terms of total net sales, with a history spanning over centuries. The total net sales of Posti Group in 2019 exceeded EUR 1564 million with an operating profit of EUR 39 million. Posti Group is owned by the state of Finland and employs approximately 22 000 staff members.

Posti's business consists of delivery services for letter, publication and parcel products, e-commerce services, supply chain solutions as well as a range of transport services for different businesses and organizations. Posti Group maintains a daily supply chain that services approximately 2.8 million households by delivering an average volume of 8 million shipments each day. Posti operates in 10 countries focusing on operations in Finland, Russia, and the Baltics [18].

eCommerce activities are growing very rapidly both globally and in Finland according to Statista [20]. Global online sales are forecasted to reach 4.5 trillion U.S. dollars in 2021 based on data collected from eMarketer [9]. While Finland lags behind in growth when compared to for example Sweden, the rate of growth is still very high. The growth in eCommerce volume in Finland is estimated to be between 7% and 10% annually for the upcoming years [20]. The popularity of online shopping seems to stem from the ease and comfort that it offers, prompting more and more people to cut down on shop visits and purchase online instead [10].

Reacting to this emerging trend, Posti is currently undergoing its greatest transformation to date. Consumer behavior has changed, digitalization is increasing, the volumes of paper mail are decreasing at a growing rate and consequently Posti is looking for significant growth opportunities in parcels, e-commerce and logistics services as a response [17]. Posti Group forecasts that with the continuing decline in volumes of traditional postal services and with a continuous increase in volume of parcel shipments and continuing growth of international and domestic eCommerce business, the Parcel and eCommerce business segment will be key in the future years. Thus, a timely and cost effective supply chain for parcel and eCommerce items is of great importance for Posti Group.

1.3 Objectives

The main objective of our project is to develop a routing and scheduling planning optimization tool that integrates the associated parcel transportation and sorting costs within a given planning period. The Posti parcel supply chain is heavily time constrained as shipments need to be delivered to the customers within a time frame of 24h and the supply network covers the whole of Finland. Furthermore, the network is entirely location based. This means that the locations of distribution hubs and distances between all network locations are fixed. These properties require that the tool being developed integrates both the routing of transport units and the scheduling of sorting facilities and transportation.

The development of the tool is based on the main tasks Posti presented for the project. Firstly, the project should study the relevant literature for finding best solution approaches to the given optimization problem considering factors such as admissibility, efficiency, scalability among others. Secondly, the project should implement a problem-solving algorithm that defines a cost function to the problem and may be tested with data provided by Posti. The results given with the algorithm are to be reported. Additionally, exploring possible extensions to the model were given as bonus tasks from Posti. These bonus tasks included investigating scenarios where transport unit trailers could be swapped between trucks using mid-way depots and improving the supply chain through optimizing the assignment of terminals under the different distribution centers.

This project work develops a Mixed-Integer-Linear Programming (MILP) model for optimizing parcel transportation and sorting within a 24 hour window such that the integrated costs are minimized. Chapter 2 presents a review of relevant literature about routing and scheduling problems. Chapter 3 presents the MILP model in detail and its implementation. In Chapter 4 we discuss the testing of the model based on data provided by Posti and we report the results. We discuss these results within the context of the original problem setting provided by Posti Group at the onset of the project in Chapter 5 and finally conclude our report in Chapter 6.

Chapter 2
Literature Review

Supply chain management optimization problems in logistics are often well suited for mathematical programming. In particular, problems in the selection of optimal transportation and manufacturing options within a finite supply chain network can often be solved with combinatorial optimization[24].

The travelling salesman problem is a widely researched NP-hard combinatorial optimization problem that attempts to find the shortest possible loop that visits each node of a network with known distances between all pairs such that the loop begins and ends in the same particular node. Most route and schedule optimization problems in logistics are built on the textbook travelling salesman problem and its generalizations.

The travelling salesman problem is NP-hard in its computational complexity. An integer programming formulation for it was first introduced by Dantzig et. al [7]. The truck dispatching problem, also presented by Dantzig et al. [8], is a specific extension of the travelling salesman model attempting to service nodes in a network with several travelling salesmen, or trucks in this particular application, operating from a single node. Clarke and Wright [6] later generalized this problem to a linear optimization problem that is commonly encountered in the domain of logistics and transport even today: how to serve a set of customers that are geographically distributed around a central depot, using a fleet of trucks with varying capacities. This setting became known as the "Vehicle Routing Problem" (VRP). The vehicle routing problem attempts to solve for an optimal routing of a fleet of vehicles starting from a single depot such that all nodes in a network are being serviced while costs are being minimized. These costs can be interpreted as service time or monetary costs of transportation for example. Based on the travelling salesman and the truck dispatching problem, the vehicle routing problem is also NP-hard. Several extensions to the vehicle routing problem have been introduced in the literature including problems with several depots, multiple routes or constrained time windows. See for example De Jaegere [1] for a review of these variants and a listing of corresponding scientific articles.

Vehicle scheduling problem is another variant of optimization problems in logistics and transportation. This problem often arises in public transportation where a fleet of buses is required to service a timetabled set of routes while minimizing transportation costs. For an overview of vehicle scheduling models in the literature see for example Bunte et al [4]. Combining vehicle or manufacturing scheduling and vehicle routing is another extension of the presented problem sets. Approaches for solving these kinds of problems are discussed by Chen et al. [5] and Ullrich [23] for example.

Generali textbook overviews of vehicle routing and scheduling problems in logistics and supply chain planning are given for example in [24], [11] or any other standard textbook of optimization in supply chain planning.

2.1 Scheduling and Vehicle Routing Problems with Time Windows

When a vehicle routing problem is extended by introducing constraints that involve time in the form of final service times, limited time windows and so on, the corresponding models are referred to as vehicle routing problems with time windows. For a fairly recent listing of literature on time constrained vehicle routing problems, see for example Bräysy and Gendreau ([2] and [3]) or Toth and Vigo [22]. Similarly to the problems that were not constrained with time, vehicle scheduling problems or vehicle routing problems with time windows combine together to form a new "family" of logistics optimization problems that attempt to optimize the routes and timings of the service fleet as a whole. An overview of solution methods for these vehicle scheduling and routing problems with time windows is presented by Solomon [19]. Solomon argues that practical sized problems of vehicle scheduling and routing using time windows require approximation methods built on various heuristics. The algorithms he proposes are used also in more recent literature as a benchmark.

For example, in [15], Lau et al. present a heuristic algorithm for the pick up and delivery problem with time windows. This problem concerns the transportation of goods from multiple source locations to multiple destinations when transport capacity and time windows are constrained which is a generalization to the standard vehicle routing problem with time windows. While being an improvement to Solomon, the approach suggested by Lau also uses a heuristic for approximating a solution instead of solving to optimality. In their approach an approximate initial solution is created through a constructive heuristic and then local search is used to refine the optimality of the initial solution. Similarly, Thangiah et al. [21] use and compare several heuristics to solve for the NP-complete case of servicing each customer of a transportation network such that a strict deadline to deliveries is observed while fleet size is also being minimized.

Indeed, using various heuristics for solving vehicle routing problems is a common approach discussed in vehicle scheduling and routing related literature [1]. Often the NP-hardness of the vehicle routing and scheduling problems makes solving them to optimality possible only on small problem instances without risking infeasible calculation times. Thus, approximations through heuristics appear to be the most commonly adopted approach for making larger problem instances solvable within acceptable timeframes. Several exact algorithms have however been proposed by Kallehauge in [14]. These exact algorithms are based on research concerning the standard travelling salesman problem. The travelling salesman problem formulations together with the vehicle routing problem formulations are based on the work of Dantzig et al.[7] and approaches for them are often based on principles of integer linear programming and different cutting plane methods. Adopting the standard (integer) linear programming makes it possible to find an optimal solution when the problem instances are of decent size. This means that the choice of exact and heuristic methods is a choice between computation time and solution quality based on the complexity and size of the underlying problem. Indeed, mixed-integer LPs have previously been used to good effect in postal services context, when internal transports have been optimized in a parcel sorting center (see [25]).

Chapter 3
Data & Methods

We model our optimization problem using a mixed integer linear programming (MILP) formulation. We assume that only a single type of parcels is being handled inside our network in the model. Also we model only one possible type of a transportation truck with fixed capacity and we do not limit how many trucks are available. Additionally, we assume that the trucks do not have to return to their starting locations, and that there is no time spent in loading and unloading of the trucks. Furthermore, we assume that loading and unloading of the trucks does not incur separate costs but all transport related costs are combined. This combined cost is dependent only on the transport distance between locations. For modeling purposes, we have discretized the 24 hour daily time window into 15 minute intervals. Other assumptions are parametric and may be varied.

The model is based around the following information. In Finland, the parcels are handled in 6 distribution centers (Vantaa, Tampere, Lieto, Kuopio, Oulu, Seinäjoki) and in 33 terminals. Each terminal is linked to one distribution center based on regional areas. When a customer sends a parcel, it is collected and sent to the corresponding distribution center which does the outbound sorting. The sorting can be done either manually or by using a sorting machine which is cheaper but has certain constraints that are introduced in later sections. After the outbound sorting, the parcel is transported to another distribution center which provides the area which is the final destination of the parcel. In that center, the parcel goes through the inbound sorting. Finally, the parcel is transported to the correct terminal to reach its receiver. This sequence is visualized in Figure 3.1.

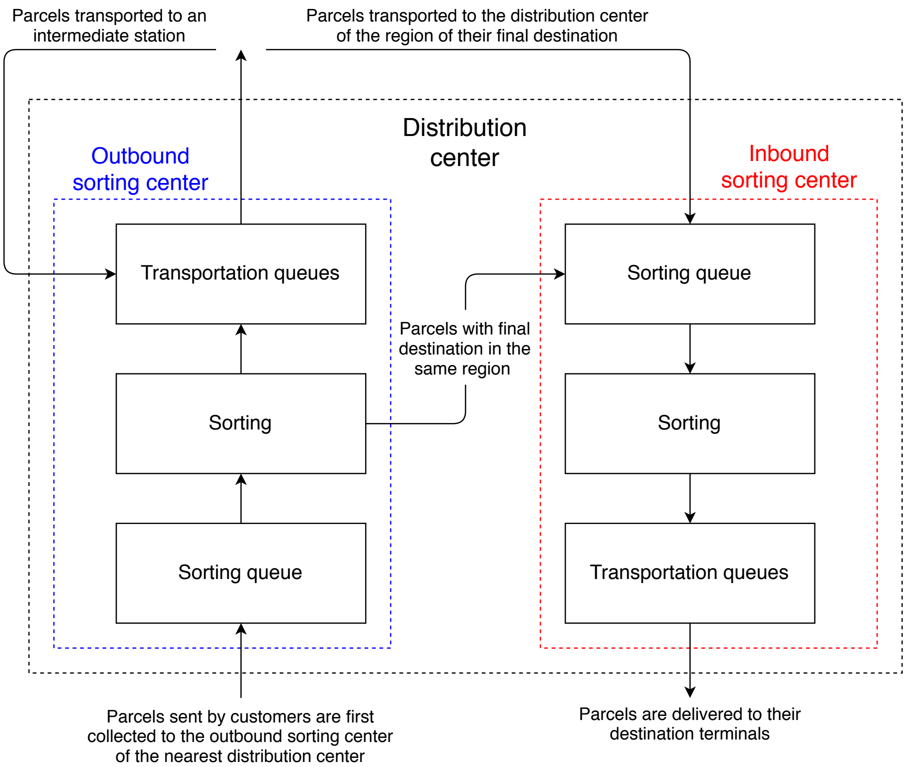

Figure 3.1: Sorting and transportation visualized with respect to one of the distribution centers.

The primary target of the model is to determine a transportation routing and scheduling plan for a 24 hour period while minimizing the cost of sorting and transportation of parcels. For sorting, this translates to minimizing the use of manual sorting and utilising the full capacity of sorting machines. For transportation, this translates to using as few trucks as possible, choosing the shortest routes, and maximizing truck loads. Furthermore, one goal regarding the routing is to assess whether utilizing intermediate stations in transportation of parcels from distribution center to another is sensible or even optimal. The main constraint for the model is that 95% of the parcels must reach their final destination terminal before the 24 hour period is over. The model is formulated as a MILP problem.

3.1 Parameters and decision variables of the MILP

The model parameters are in Table 3.1. The table is divided into two parts. Parameters in the upper part are fixed parameters including information about the locations, capacities and costs. Parameters in the lower part are more daily dependent: how many parcels enter the network and what is the final destination of these parcels. These are estimated from a raw data which includes the origin and final destination of every parcel during a 24 hour period.

Table 3.1: Parameters of the model

Formula	Description	Unit
$t \in T = \{1, ..., 96\}$	24 hour period divided into 15 minute intervals	
$i, j, k \in J = \{1, ..., 6\}$	indices of the distribution centers	
$l \in L = \{1, ..., 33\}$	indices of the terminals	
$L_i \subset L$	the set of terminals under the region of distribution center i	
T^c_{ij}	traveling time between two distribution centers	0.25h
T^t_{il}	traveling time between a distribution center and a terminal	0.25h
A^{cap}_i	machine sorting capacity of center i during 15 minutes	parcels
M_{cap}	manual labor capacity per worker during 15 minutes	parcels/worker
M_{lim}	available manual labor workers for sorting	worker
C^a_i	cost of machine sorting per 15 minutes in center i	euros
C^m	cost of manual sorting per 15 minutes per worker	euros/worker
C^t	cost of transportation unit per 15 minutes	euros/0.25h/truck
C^l	penalty cost for late delivery for a parcel	euros/parcel
T_{cap}	parcel capacity of one transportation	parcels/truck
s_{ti}	the number of parcels arriving in outbound sorting center i at time t from terminals	parcels
G_{ik}	proportion of parcels sorted in center i going to center k	
G^t_{il}	proportion of parcels sorted in center i going to terminal l, $G^t_{il} = 0$ if $l \notin L_i$	

The model includes only non-negative variables. The list of variables is the following one:

- continuous decision variables
  - a^{out}_{ti}, a^{in}_{ti}: the amount of outbound (out) and inbound (in) machine sorting done during 15 minutes in center $i \in J$ at time $t \in T$ (measured in parcels)
  - m^{out}_{ti}, m^{in}_{ti}: the amount of outbound and inbound manual sorting done during 15 minutes in center $i \in J$ at time $t \in T$ (measured in parcels)
  - x_{tijk}: the number of parcels transported from center $i \in J$ to $j \in J/\{i\}$ when the transportation starts at time $t \in T$ and the parcels have a final destination $k \in J/\{i\}$. The case $j = k$ is interpreted as a direct transportation from i to k
  - b_{til}: the number of parcels transported from center $i \in J$ to terminal $l \in L_i$ when the transportation starts at time $t \in T$,
- binary decision variables
  - z^{out}_{ti}, z^{in}_{ti}: indicates whether the machine is available (1) or not (0) for outbound and inbound sorting in sorting center $i \in J$ at time $t \in T$,
- integer decision variables
  - y_{tij}: number of trucks sent from center $i \in J$ to center $j \in J/\{i\}$ at time $t \in T$
  - y_{til}: number of trucks sent from center $i \in J$ to terminal $l \in L_i$ at time $t \in T$,
- additional variables
  - A_{ti}: the number of parcels waiting for outbound sorting in center $i \in J$ at the start of time interval $t \in T \cup \{97\}$

- B_{tik}: the number of parcels waiting for transportation from center $i \in J$ to center $k \in J/\{i\}$ at the start of time interval $t \in T \cup \{97\}$
- C_{ti}: the amount of parcels waiting for inbound sorting in center $i \in J$ at the start of time interval $t \in T \cup \{97\}$
- D_{til}: the amount of parcels waiting for transportation from center $i \in J$ to terminal $l \in L_i$ at the start of time interval $t \in T \cup \{97\}$.

3.2 Objective function

The objective function to be minimized is the sum of the sorting costs (3.1), transportation costs (3.2) and late delivery penalty costs (3.3) over the time period of 24 hours. Sorting cost is the sum of costs of machine sorting and manual sorting. Cost of machine sorting is constant while the machine is running, and zero when it is not. Cost of manual sorting is proportional to the number of parcels being sorted. Transportation cost is a function of driving distances and number of truck used. Late delivery cost is the number of parcels that are not delivered in time multiplied by a unit cost.

Sorting cost:

$$\sum_{t \in T} \sum_{i \in J} \{(z_{ti}^{out} + z_{ti}^{in})C_i^a + (m_{ti}^{out} + m_{ti}^{in})\frac{C^m}{M_{cap}}\} \tag{3.1}$$

Transportation cost:

$$C^t \sum_{t \in T} \sum_{i \in J} \{ \sum_{j \in J/\{i\}} y_{tij} T_{ij}^c + \sum_{l \in L_i} y_{til} T_{il}^t \} \tag{3.2}$$

Late delivery penalty cost:

$$C^l (\sum_{t \in T} \sum_{i \in J} s_{ti} - \sum_{i \in J} \sum_{l \in L_i} \sum_{t=1}^{96-T_{il}^t} b_{til}) \tag{3.3}$$

3.3 Constraints

$$\sum_{k \in J/\{i\}} x_{tijk} \leq y_{tij} T_{cap} \quad \forall t \in T, i \in J, j \in J/\{i\} \tag{3.4}$$

$$b_{til} \leq y_{til} T_{cap} \quad \forall t \in T, i \in J, l \in L_i \tag{3.5}$$

$$\sum_{j \in J/\{i\}} x_{tijk} \leq B_{tik} \quad \forall t \in T, i \in J, k \in J/\{i\} \tag{3.6}$$

$$b_{til} \leq D_{til} \quad \forall t \in T, i \in J, l \in L_i \tag{3.7}$$

$$a_{ti}^{out} + m_{ti}^{out} \leq A_{ti} \quad \forall t \in T, i \in J \tag{3.8}$$

$$a_{ti}^{in} + m_{ti}^{in} \leq C_{ti} \quad \forall t \in T, i \in J \tag{3.9}$$

$$\sum_{t \in T} (a_{ti}^{out} + m_{ti}^{out}) = \sum_{t \in T} s_{ti} \quad \forall i \in J \tag{3.10}$$

$$a_{ti}^{out} \leq z_{ti}^{out} A_i^{cap} \quad \forall t \in T, i \in J \tag{3.11}$$

$$a_{ti}^{in} \leq z_{ti}^{in} A_i^{cap} \quad \forall t \in T, i \in J \tag{3.12}$$

$$z_{ti}^{out} + z_{ti}^{in} \leq 1 \quad \forall t \in T, i \in J \tag{3.13}$$

$$m_{ti}^{out} \leq M_{cap} M_{lim} \quad \forall t \in T, i \in J \tag{3.14}$$

$$m_{ti}^{in} \leq M_{cap} M_{lim} \quad \forall t \in T, i \in J \tag{3.15}$$

$$A_{(t+1)i} = A_{ti} + s_{ti} - a_{ti}^{out} - m_{ti}^{out} \quad \forall t \in T, i \in J \tag{3.16}$$

$$B_{(t+1)ik} = B_{tik} + G_{ik}(a_{ti}^{out} + m_{ti}^{out}) - \sum_{j \in J/\{i\}} x_{tijk} + \sum_{j \in J/\{i,k\},\, t - T_{ji}^c \geq 1} x_{(t-T_{ji}^c)jik} \quad \forall t \in T, i \in J, k \in J/\{i\} \tag{3.17}$$

$$C_{(t+1)i} = C_{ti} + G_{ii}(a_{ti}^{out} + m_{ti}^{out}) - a_{ti}^{in} - m_{ti}^{in} + \sum_{j \in J/\{i\},\, t - T_{ji}^c \geq 1} x_{(t-T_{ji}^c)jii} \quad \forall t \in T, i \in J \tag{3.18}$$

$$D_{(t+1)il} = D_{til} + G_{il}^t(a_{ti}^{in} + m_{ti}^{in}) - b_{til} \quad \forall t \in T, i \in J, l \in L_i \tag{3.19}$$

$$A_{(t=1)i} = 0 \quad \forall i \in J \tag{3.20}$$

$$B_{(t=1)ik} = 0 \quad \forall i \in J, k \in J/\{i\} \tag{3.21}$$

$$C_{(t=1)i} = 0 \quad \forall i \in J \tag{3.22}$$

$$D_{(t=1)il} = 0 \quad \forall i \in J, l \in L_i \tag{3.23}$$

$$\sum_{i \in J} \sum_{l \in L_i} \sum_{t=1}^{96 - T_{il}^t} b_{til} \geq 0.95 \sum_{t \in T} \sum_{i \in J} s_{ti} \tag{3.24}$$

The purposes of these constraints are as follows. Constraint (3.4) and (3.5) ensure that we can only transport as many parcels as we can fit in trucks assigned for the given route. Constraints (3.6) and (3.7) ensure that parcels cannot be transported without first sorting them. Constraints (3.8) and (3.9) ensure that you can only sort parcels that are waiting for being sorted. Constraint (3.10) ensures that every parcel goes through outbound sorting. Constraints (3.11)-(3.13) ensure that we can only use machine sorting according to its capacity and only do either outbound or inbound sorting at a time. Constraints (3.14) and (3.15) ensure that we can only use manual sorting according to available workers. Constraints (3.16)-(3.19) are the balance constraints for the queues. Constraints (3.20)-(3.23) state that the queues are initially empty. Constraint (3.24) is the obligation to deliver at least 95% of the parcels in time.

In Figure 3.2, the problem is represented using the aforementioned parameters, variables and constraints.

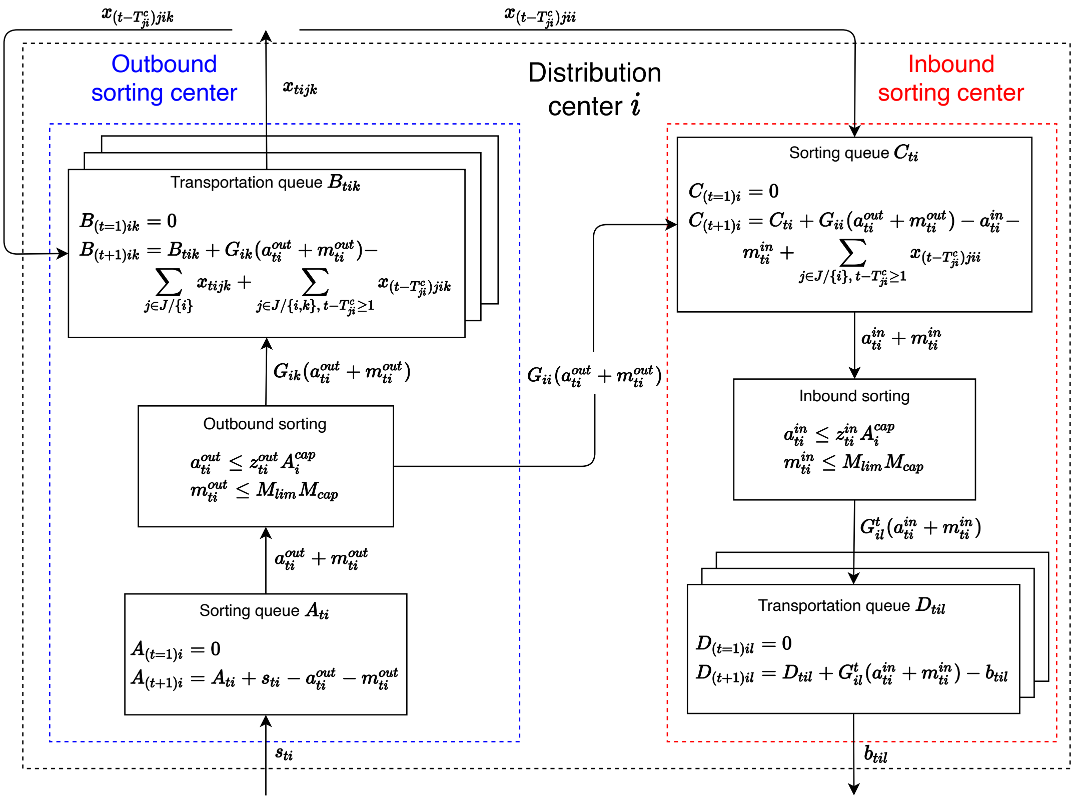

Figure 3.2: More detailed chart of the problem.

3.4 Description of the route model

In addition to some smaller variations to the base model, we also experiment with the so-called route model. In this variant only a subset of routes is allowed for routing the parcels. In addition to simplifying the optimized solutions, this allows us to remove some variables from the model which could potentially speed up the optimization process.

In practice, this change is implemented by first defining for each pair of sorting centers

$(i, j) \in J \times J$ the set $R_{ij} \subset J$ that lists the final destinations (distribution centers) that parcels that are sent from distribution center i to j may have. As an example, assume that we have the set $R_{1,2} = \{2, 5\}$. This means that a truck leaving from Vantaa (1) to Tampere (2) can only contain parcels that have a final destination in either Tampere or Oulu (5). It is worth noting that, if it also holds that $5 \in R_{2,6}$, it is possible that a parcel that is headed from Vantaa to Oulu is first transported from Vantaa to Tampere, then to Seinäjoki (6), and finally to Oulu. That is, this definition enables also routes with more than one intermediate center.

With this definition we can then modify the model so that transportation variables x_{tijk} are only created for indices i, j, k that satisfy $k \in R_{ij}$. Consequently, the sums in constraints (3.4), (3.6), (3.17), and (3.18) are changed to

$$\sum_{k \in R_{ij}} x_{tijk} \ ,$$

$$\sum_{j \in J \ | \ k \in R_{ij}} x_{tijk} \ ,$$

$$-\sum_{j \in J \ | \ k \in R_{ij}} x_{tijk} + \sum_{\substack{j \in J \ | \ k \in R_{ji}, \\ t - T^c_{ji} \geq 1}} x_{(t - T^c_{ji})jik} \ , \text{and}$$

$$\sum_{\substack{j \in J \ | \ i \in R_{ji}, \\ t - T^c_{ji} \geq 1}} x_{(t - T^c_{ji})jii} \ ,$$

respectively.

3.5 Solution method and validation

The model is implemented using Julia and JuMP (Julia for Mathematical Programming). It is solved using the Gurobi solver. Gurobi first solves the model by disregarding the integer constraints of the variables (linear relaxation). Then it uses different solving methods in order to find feasible integer solutions [13]. The goodness of each integer solution is determined with an optimality gap which is the difference between the objective values of an integer solution and a linearly relaxed solution.

To validate the model we used dummy parameter values. We set s_{ti} such that only a small number of parcels entered the network so we could keep track of the results easily. We also set G_{ik} and G^t_{il} such that parcels were required to be sent everywhere in Finland. This validation gave us good insight in to the model behaviour. The validation indicated that our model was behaving reasonably well and as expected. All parcels were machine sorted which is the cheaper option of the available sorting methods. Also the model did not suggest any unreasonable transports.

Validation also gave us other insights for model implementation. At first, variables x_{tijk} and b_{til} were defined for all centers $i, j, k \in J$ and terminals $l \in L$ which meant that additional constraints forcing certain variable values to zero had to be included to avoid, for example, transportations from center to itself and transportations from Vantaa to a terminal linked to Oulu. In the current definitions in section 3.1 we do not initialize these variables that would be forced to zero, reducing the number of variables and constraints in the model. This helped us reduce computation time.

3.6 Data flow and parameter values

Values for most of the model parameters are set in the beginning of code according to information from Posti and are modifiable at run-time. This includes the linking of terminals to sorting centers, which is based on a file combining zip codes (postal codes) to a terminal and a sorting center that Posti has provided. To modify this, additional information on the driving time between the terminal and the new sorting center is required. The raw values of driving times, set manually in the code, are rounded up to 15 minute precision, and 45 minutes is added to all driving times exceeding 4.5 hours to address the break required for drivers by law. Standard value for the late delivery cost is $C^l = 5$, based on our own tests (see Section 4.3.3). This is the value used in all other tests described in this report.

The parameters s, G, and G^t are read in from csv-files during execution, using a format described in more detail in the code. The values used in the tests are based on data from Posti that contained information on shipments handled in outbound sorting during one week. For each shipment the data contained date, time (15 minute precision), departure and destination zip codes, and the number of parcels.

To define the inflow of parcels s, the time stamps in the data were interpreted as the time of arrival to the outbound sorting center. The shipments were grouped by date, time interval, and outbound sorting center (defined by the departure zip code). The numbers of parcels were then summed up per group, providing us with seven data sets (one per day of the week) in a form acceptable to the Julia code. In the tests only data from the day with the highest number of incoming parcels was used. In the code this data is further modified so that at time $t = 1$ we have the sum of values from time intervals $t \in [1, 33]$, followed by the remaining 63 time intervals, and the values of s_{ti} for all $t \geq 65$ are zeros. This way the 24 hour period in the model starts at 8.00 in the morning with the arrival of all parcels between 0.00 and 8.00, followed by the unmodified inflow data until 23.45 on that same day. This way all parcels in the original data are considered, but those arriving after midnight only need to be delivered during the next 24 hour period (that is, by 8.00 on the day after they arrived to outbound sorting).

When defining G, the shipments were grouped by date, inbound sorting center, and outbound sorting center. The numbers of parcels per group were summed up and divided by the total by date and outbound sorting center, yielding proportions of parcels going from each outbound sorting center to each inbound sorting center, one set per day of the week. To define G^t the shipments were grouped by date and terminal, summed up by groups, and divided by the daily total of the inbound sorting center corresponding to each terminal. Additionally, these proportions had to be ordered according to the list in the code. As is visible already from the model definition, the parameters G and G^t are constant during the whole optimization period, meaning that whenever any parcels are sorted a proportion of them is deemed to have final destination in each of the other sorting centers (for G), or of the terminals linked to the sorting center (for G^t), assuming that proportion wasn't zero during the entire day in the source data.

In addition to the output printed by the Gurobi optimizer as a result of the optimization process, the code prints out relevant information related to the solution. These include the departure times, start and end points, and contents (parcels by final destination) of all transportations, as well information on all sorting activities (time, machine/manual, direction, number of parcels). This info is also written into csv-files, and figures that are drawn visualize the solution are stored in pdf-files. The contents and formats of the outputs are described in more detail in the code.

Chapter 4
Results

This Chapter presents different model results based on the data provided by Posti. We ran the model using several computers with varying performances. In detail, we used the following machines:

- In Section 4.1 we had a laptop, that has 2.3 GHz dual-core Intel Core i5 processor, 8 GB of RAM and an operating system of macOS
- In Section 4.2 we had a laptop with a 2.1 GHz Intel Core i3-5010U CPU using 2 compute cores with 4 GB of RAM and operating system 64-bit Windows 10
- In Section 4.3.2 we had a laptop with 2.40GHz AMD A6-9210 RADEON R4 processor using 2 compute cores with 8GB of RAM and operating system of Windows 10.
- In Section 4.3.3 we had a desktop with 3.50GHz Intel i5-4690K processor using 4 compute cores with 8GB of RAM and operating system of Windows 10.

Every time the model was run the optimization time was limited to 1000s.

4.1 Results for the baseline model

The baseline model was introduced in the previous chapter. In this section, the results that were obtained with the model are presented. In particular, solution costs, sorting, terminal transportations and routes used in transportations between distribution centers are analyzed. Furthermore, the methods implemented for analyzing the results and saving the results for further studying are introduced.

Optimization resulted in a solution which involved a total cost of 117990.38 with 8.45 % optimality gap. 66595.80 of the total cost was due to sorting, 20734.00 was due to transportation and 30660.58 was due to late delivery penalties. Furthermore, 97.1 % of parcels were delivered in time. In itself, these values do not provide enough information about the solution. Further analysis of the variables is needed.

The sorting and transportation queues of all distribution centers were analysed together with the subject matter experts at Posti to validate the baseline results. In Figure 4.1, the sorting machine usage is plotted. The horizontal axis represents time. Time varies between 1 and 97. In the model, 24 hours is divided to 15 minute sections which correspond to 96 time periods. In addition, queue-variables require one additional time point because of the constraints (3.16)-(3.19), thus 97 time points in total. The figure shows that outbound sorting is done earlier than inbound sorting. This is in line with the network structure: parcels first arrive to the outbound sorting queue and later to the inbound sorting queue. Further analysis also showed that parcels are mostly sorted with a sorting machine, the sorting machine capacity is utilised efficiently and, most importantly, the parcels in the network are being sorted.

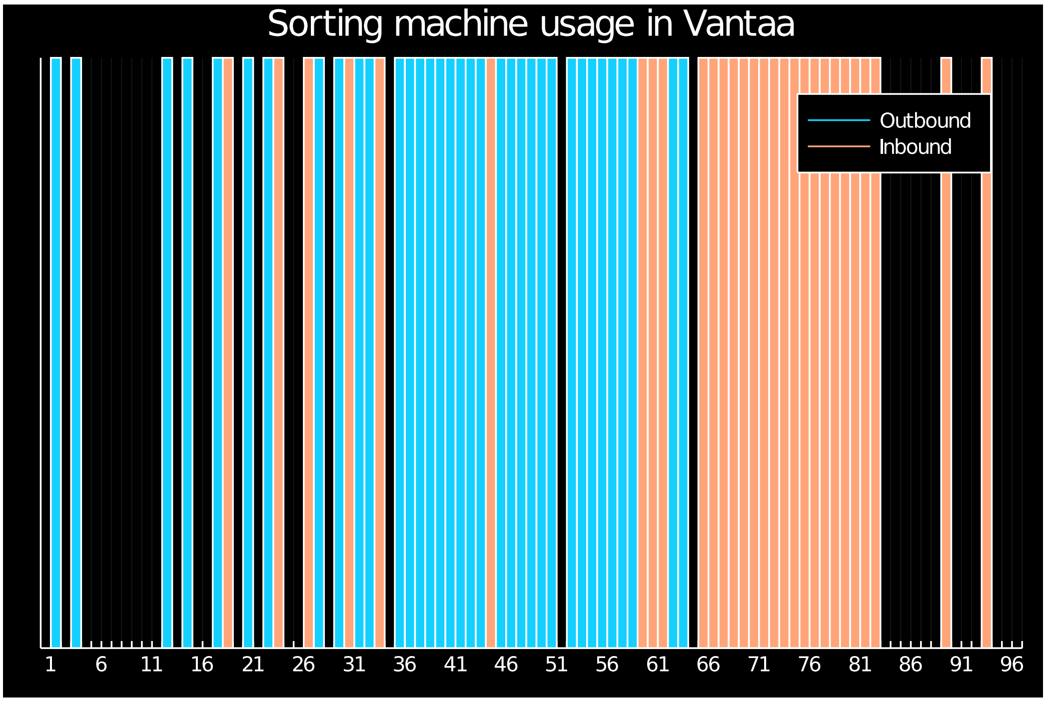

Figure 4.1: Sorting machine usage in Vantaa. Horizontal axis represents time.

Figures 4.2, 4.3 and 4.4 show three transportation queues (yellow bars) and the respective transportations (blue bars). In Figures 4.2 and 4.3, capacities of the transportation units (TU) are utilised efficiently: there is minimum amount of empty space (pink bars) in the TUs. On the other hand in Figure 4.4, clearly a sub optimal decision has been made by transporting very few parcels at a time. From Kuopio, parcels are delivered to the first terminal in time. However from Oulu, some parcels are not delivered before the deadline represented by the red vertical line. From testing the model, it has become evident that it is most challenging to deliver parcels to Oulu in time.

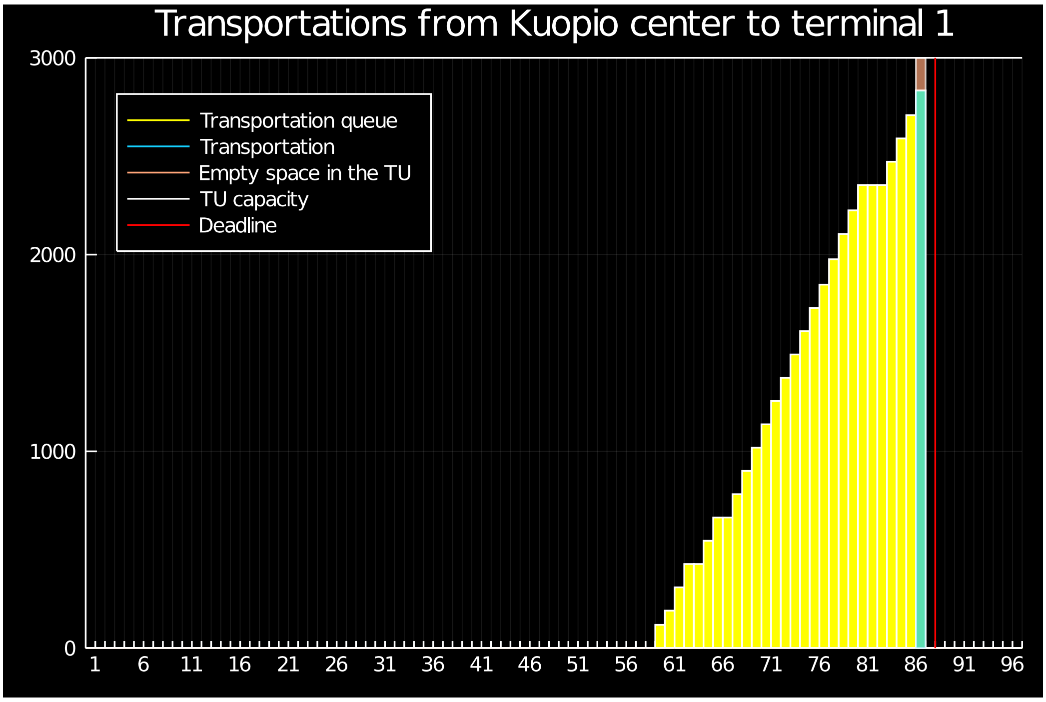

Figure 4.2: Transportations from center in Kuopio to its first terminal. Number of parcels on the vertical axis, time on the horizontal axis.

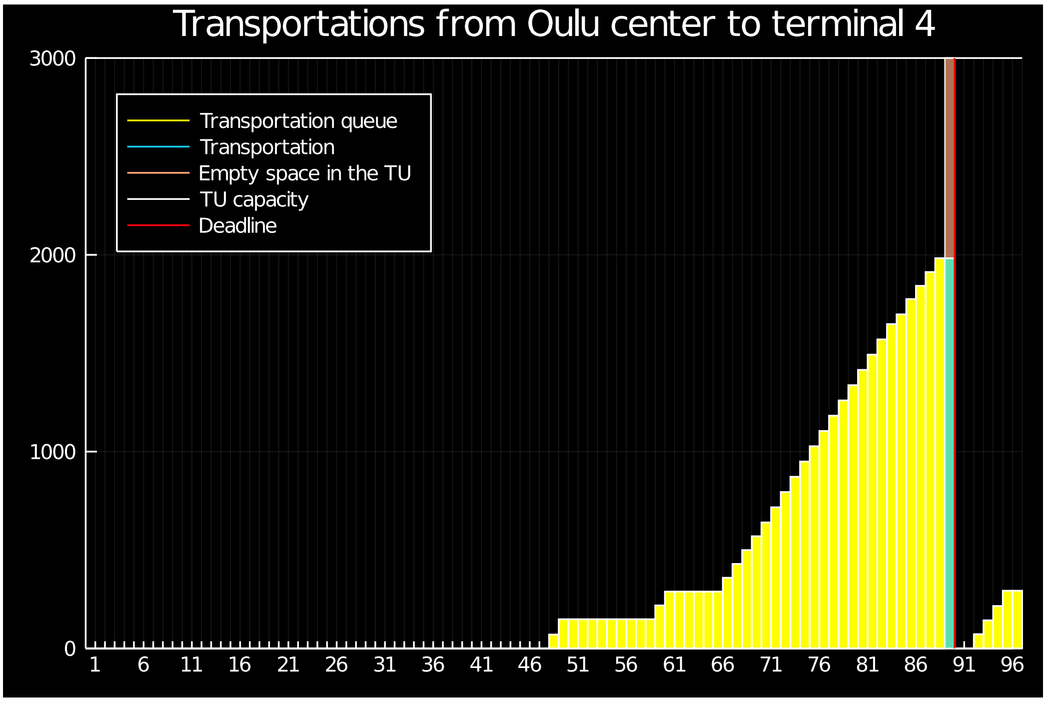

Figure 4.3: Transportations from center in Oulu to its fourth terminal. Number of parcels on the vertical axis, time on the horizontal axis.

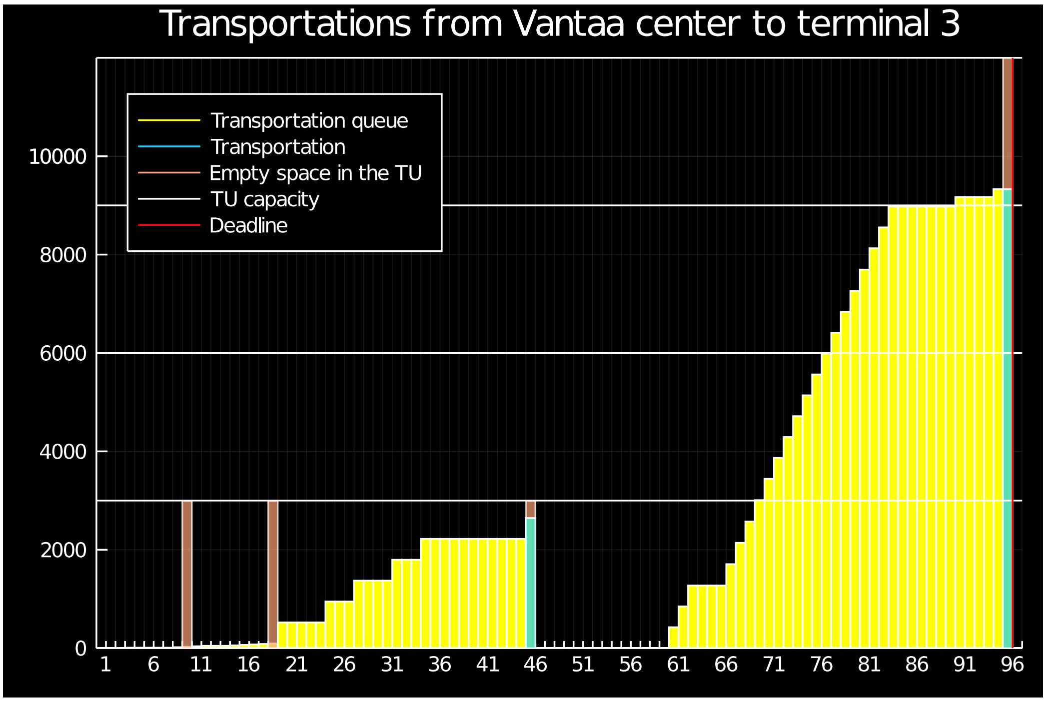

Figure 4.4: Transportations from center in Vantaa to its third terminal. Number of parcels on the vertical axis, time on the horizontal axis.

Figure 4.5 displays the routes that parcels take to the destination center. The colors of the arrows match with the final destination colors. For example, from Vantaa (left-upper figure) parcels with final destination in Oulu (blue arrows), 72 % are transported to Oulu directly and the rest are transported to Kuopio.

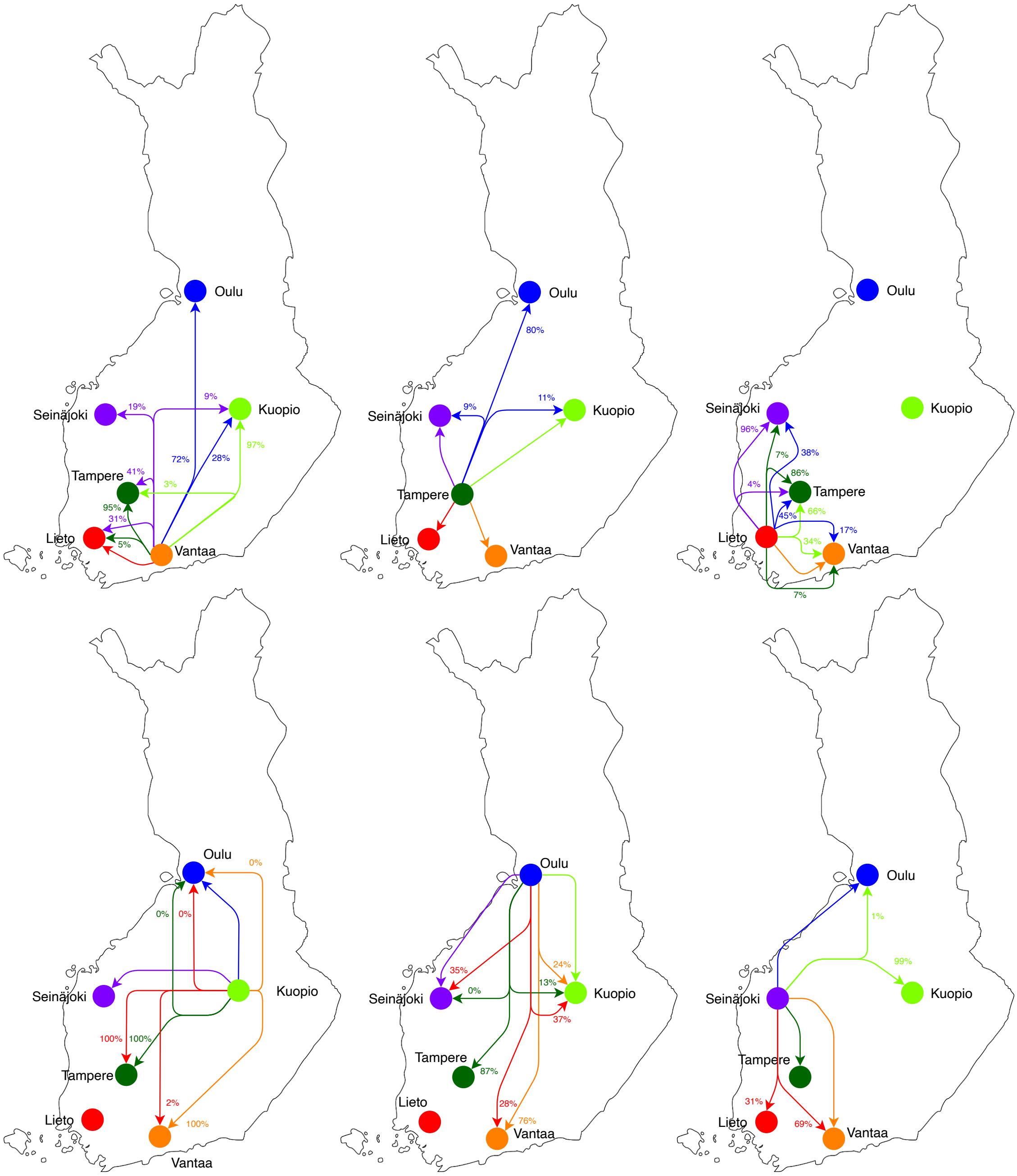

Figure 4.5: All the routes that are used in transportation between distribution centers. Proportion of parcels using the route is rounded to the closest percentage.

For the most part, the routes are reasonable. Often parcels are transported either directly to the destination center or to a center that is located some where on the way to the destination. Furthermore, there are connections, such as Lieto-Kuopio, where parcels are never transported directly to the destination but instead through an intermediate center, in this case through Vantaa or Tampere. However, there are some odd routes that are used by a very small proportion of parcels. In practice, transportation planning is not possible with such precision. With a restricted version of the model that allows only the use of predetermined routes, this type of impracticality could potentially be avoided.

Methods to produce figures as shown above have been implemented into the model code except for Figure 4.5. Nevertheless, the data providing the same information is available and can be saved to a csv-file.

4.2 Results of the route model

The route model was tested with two sets of allowed routes (”wide” and ”narrow”), shown in Table 4.1. The solutions achieved with these were compared to a run with the base model on the same machine. The main results from these tests are presented in Table 4.2. Sorting cost is not listed as it was close to equal with all versions.

Perhaps the main observation from these results is that the optimality gap increases as the selection of allowed routes becomes more limited. This suggests that the optimization in fact becomes slower or more difficult since the solver is unable to find solutions as good as with the base model even though it should be possible. This is contrary to our expectations as the number of variables is reduced by limiting the allowed routes, even if this only affects continuous variables which are the lesser issue considering computational difficulty of the model.

There is a particularly large difference in transportation cost between the route selection and the other variants, but it seems that this is indeed rather caused by failure to find a good solution than by such a solution not existing because of added limitations. A closer inspection of the routes used by the different versions revealed that the most significant differences in routing between the narrow version and the others were on routes that were allowed in all versions. Only three of the routes excluded from the narrow selection were used significantly by the other versions (and one of them was excluded also from the wide selection), but they are not relevant enough to explain the increase in transportation cost. On the other hand, the narrow version does not use the route Oulu→Seinäjoki→Lieto at all even though it is allowed, and the base and wide versions use it for 100 and 80 % of relevant parcels, respectively.

As the results were so unexpected the tests were run another time on the same machine, and additionally once on the machine used in Section 4.1. On the same machine the results were rather similar, though difference in transportation cost was not as big. On a different machine, however, the result was nearly equally good with every version, with all optimality gaps between 8.4 and 9.1 %. This is in line with the observation that reducing the number of variables as described in the end of Section 3.5 did not appear as beneficial for all machines. Whether this is just coincidence or related to properties of the machines, we were not able to tell. Either way these observations suggest that the route model should not be judged too harshly by the results presented in Table 4.1.

Table 4.1: The first six rows represent a wider selection of allowed routes. The last six rows define a more limited ("narrow") selection of routes.

	$R_{i,1}$	$R_{i,2}$	$R_{i,3}$	$R_{i,4}$	$R_{i,5}$	$R_{i,6}$
$R_{1,j}$	{}	{2,3,4,5,6}	{2,3,5,6}	{4,5}	{5}	{5,6}
$R_{2,j}$	{1,3}	{}	{1,3}	{4,5}	{5}	{4,5,6}
$R_{3,j}$	{1,2,4,5,6}	{1,2,4,5,6}	{}	{4,5}	{5}	{4,5,6}
$R_{4,j}$	{1,3}	{1,2,3,6}	{3}	{}	{5}	{2,3,6}
$R_{5,j}$	{1,3}	{1,2,3}	{1,3}	{1,2,3,4,6}	{}	{1,2,3,6}
$R_{6,j}$	{1,3}	{1,2,3,4}	{1,3}	{4,5}	{5}	{}
$R_{1,j}$	{}	{2,4,5,6}	{3}	{4,5}	{5}	{5,6}
$R_{2,j}$	{1}	{}	{3}	{4,5}	{5}	{5,6}
$R_{3,j}$	{1,4,5}	{2,4,5,6}	{}	{4,5}	{5}	{5,6}
$R_{4,j}$	{1,3}	{1,2,3}	{3}	{}	{5}	{6}
$R_{5,j}$	{1,3}	{1,2,3}	{3}	{1,2,3,4}	{}	{1,2,3,6}
$R_{6,j}$	{1}	{1,2,3}	{3}	{4}	{5}	{}

Table 4.2: Results from tests with the route model.

Allowed routes	All (baseline)	Wide	Narrow
# of continuous variables	34636	27724	25804
Transportation cost	21112.0	21672.0	30016.0
Late delivery cost	29808.8	34077.6	32870.2
Objective cost	117819.3	122785.8	130571.6
Optimality gap (%)	8.20	11.88	16.94

4.3 Other investigations

4.3.1 Widening the machine sorting window

In reality, it is not necessarily possible to change the direction of the sorting machines every 15 minutes, which is what the model might propose from time to time. Because of this, we also experimented with widening the machine sorting window from 15 minutes to 30 minutes. In practice this was implemented so that the values of binary variables z_{ti}^{out} and z_{ti}^{in} on even time intervals t must be equal to that of the corresponding variable on the previous time interval $t-1$. Thus, the variables are essentially combined into one. This change could be very interesting, because it can lower the number of integer variables in the problem, which should speed up the optimization process. However, we also ran a test where all sorting direction variables z_{ti}^{out} and z_{ti}^{in} were continuous instead of integer, and even in that case the running time had to be limited. Therefore, we can conclude that the speed-up effect of this change would not be decisive even if the implementation removed some variables altogether instead of "combining" them with others.

The effect of this change on the solution was not significant as it mainly resulted in a 2-3

% increase in the sorting and late delivery costs, in addition to removing the unwanted behavior in the solution described above. Nonetheless, the implementation of this variant was left in the model code as optional so that the user can choose the sorting window size with 15 minute precision (window size in minutes is 15, 30, 45, 60, ...).

4.3.2 Effect of transportation unit cost

Next we analyzed the sensitivity of the model to changes in the cost of transportation unit per 15 minutes C^t. We first ran the model using original cost. Then we increased C^t, first 10% then 20% and last 30%. The key results of the optimization are in Table 4.3.

Table 4.3: Effect of transport unit cost

Value of C^t	Optimality gap (%)	Transportation cost	Cost of objective function
Original	10.0	20664.0	119916.98
+10%	15.1	26796.0 (+29.7%)	128876.33 (+7.5%)
+20%	11.0	23990.4 (+16.1%)	124672.31 (+3.9%)
+30%	11.9	27791.4 (+34.5%)	127797.28 (+6.6%)

We see that using different costs creates different kind of cost differences in the transportation costs. It seems that 10% increase in C^t is worst, because it causes the optimization to end in a solution where transportation costs have increased 29.7% and the gap is also high. Conversely when C^t increases 20% a new solution is found where transportation costs have increased less.

To analyze the situation further we examined the values of transport variables x_{tijk}. Figure 4.6 presents the parcel transports from Vantaa (most left), Tampere and Lieto (most right) to other sorting centers with original value of C^t. The colors of the arrows match with the final destination color. For example, 71% of the parcels from Vantaa with final destination in Oulu are transported there directly, 26% of those are first transported to Kuopio and 3% of those go first to Tampere. Figures 4.7-4.9 have the same information but with the increased transportation cost. Figures 4.10-4.13 present the situation with respect to parcels sent from Kuopio, Oulu and Seinäjoki.

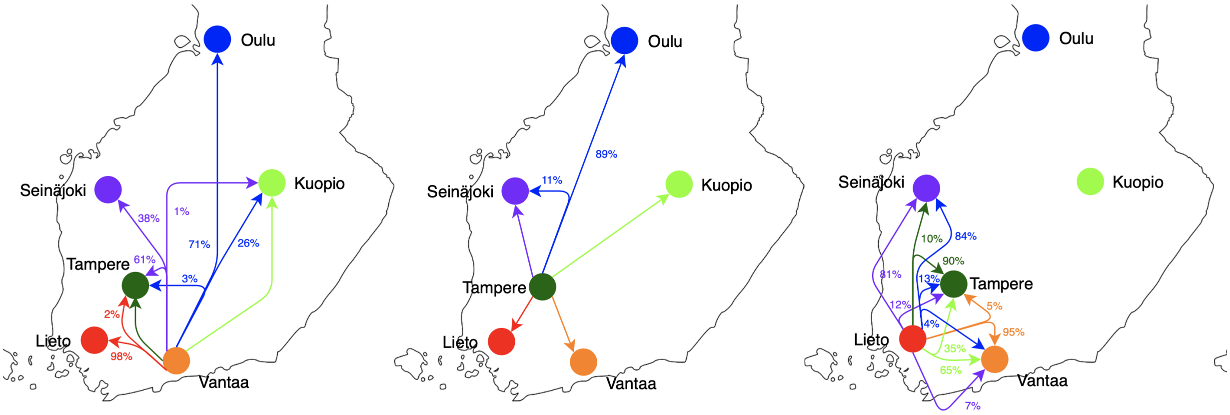

Figure 4.6: Parcels transported from Vantaa, Tampere and Lieto with original C^t

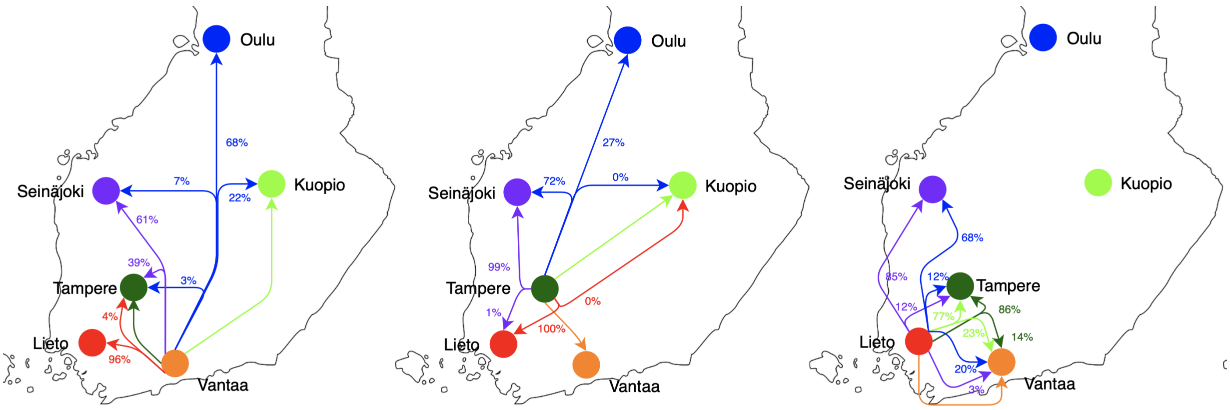

Figure 4.7: Parcels transported from Vantaa, Tampere and Lieto with 10% increase in C^t

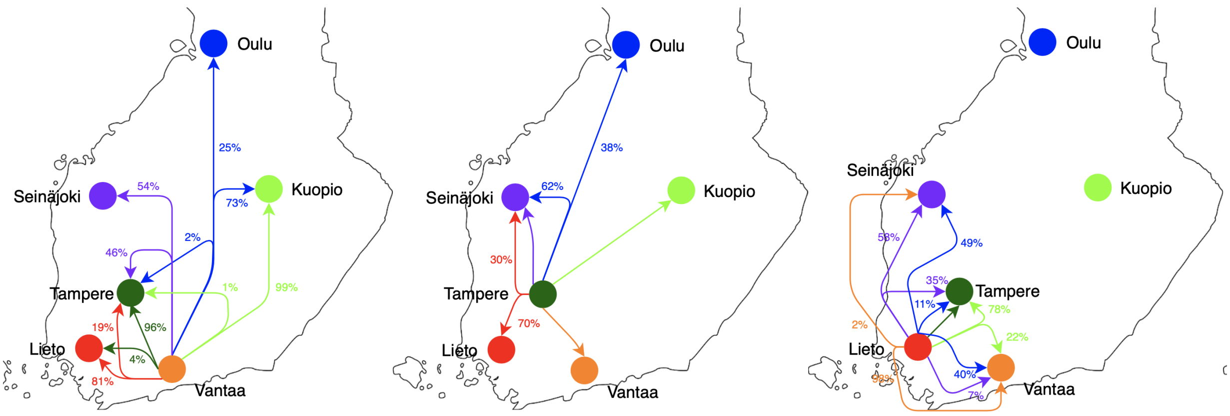

Figure 4.8: Parcels transported from Vantaa, Tampere and Lieto with 20% increase in C^t

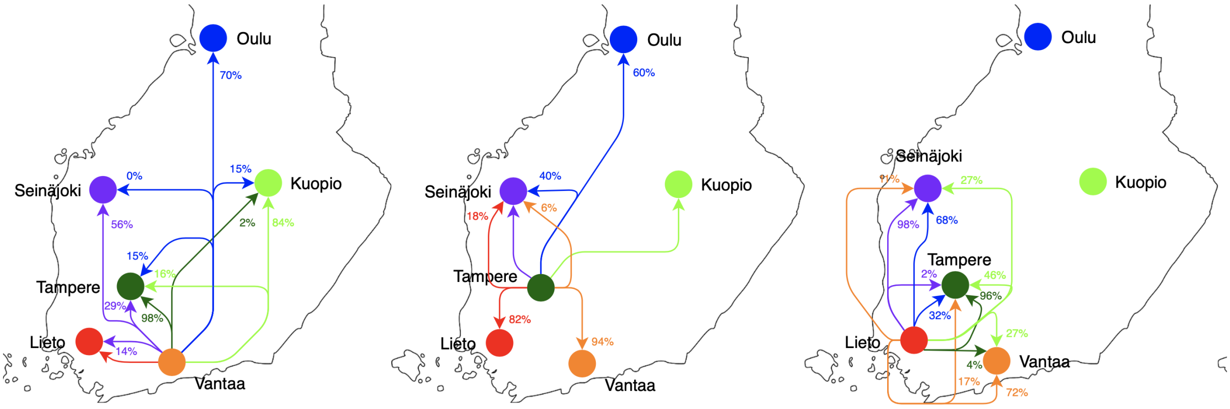

Figure 4.9: Parcels transported from Vantaa, Tampere and Lieto with 30% increase in C^t

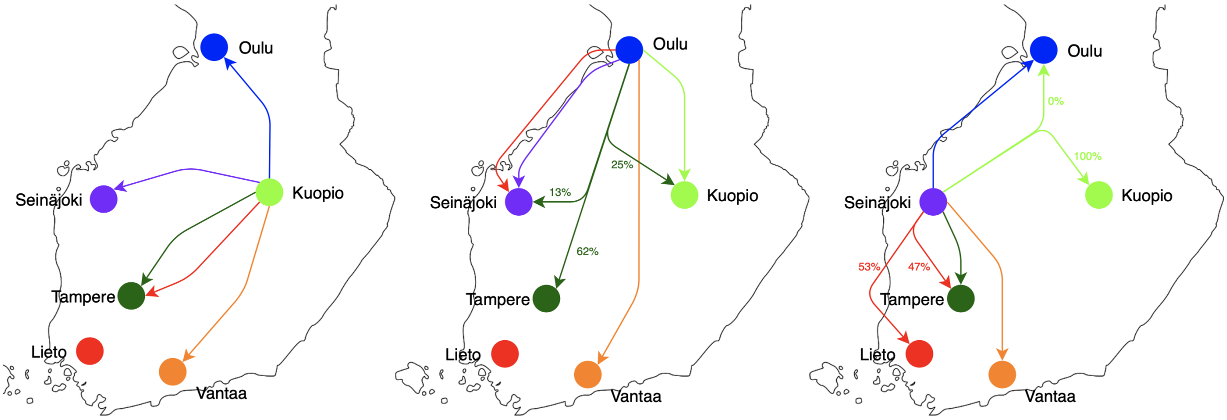

Figure 4.10: Parcels transported from Kuopio, Oulu and Seinäjoki with original C^t

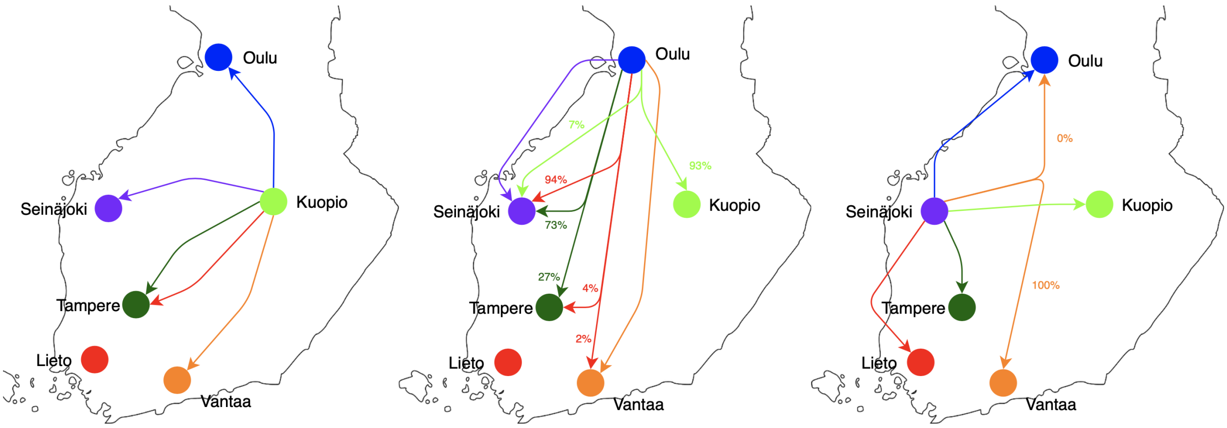

Figure 4.11: Parcels transported from Kuopio, Oulu and Seinäjoki with 10% increase in C^t

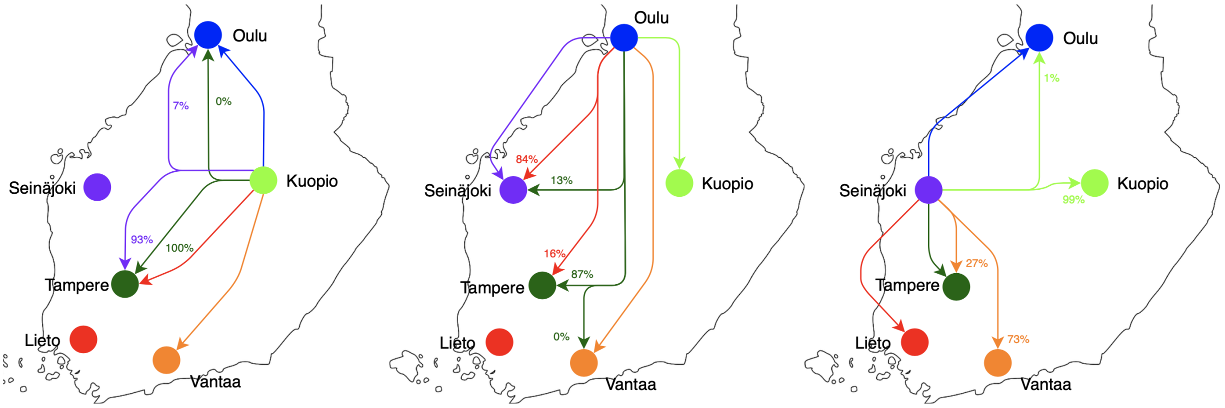

Figure 4.12: Parcels transported from Kuopio, Oulu and Seinäjoki with 20% increase in C^t

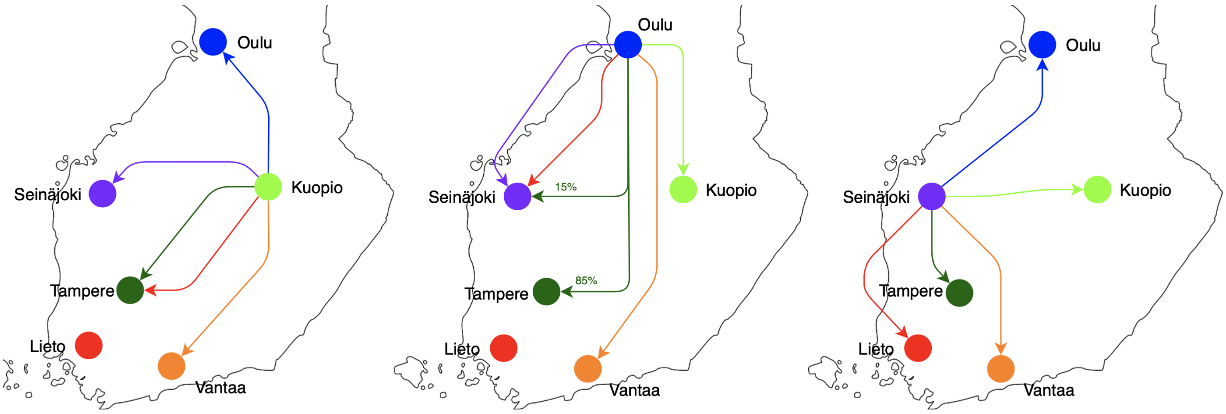

Figure 4.13: Parcels transported from Kuopio, Oulu and Seinäjoki with 30% increase in C^t

As seen in Figures 4.6-4.13 there are changes and stability with different transportation unit costs. For example, parcels from Lieto with final destination in Kuopio or Oulu are never transported to the final destination directly. Same thing happens also when parcels are moved to other direction, from Oulu and Kuopio to Lieto. This was noticed also earlier when looking at the baseline model.

However, it is hard to find clear trends when the transportation costs increase. One of them is that more parcels from Vantaa to Kuopio are sent through Tampere when transportation unit cost increases. Also parcels from Seinäjoki to Lieto are sent with a direct transport when cost increases.

To summarise, increasing the transportation unit cost did not directly increase the total cost of transportations. For example, with 20% increase, the total cost, as well as the optimality gap, were lower than with the 10% increase. Furthermore, this analysis provided some further insight regarding the transportation routes but clear interpretations about the results need practical expertise how the current Posti's network works in real life.

4.3.3 Effect of late delivery penalty cost

Next we analyzed the sensitivity of the model to changes in the late delivery penalty cost C^l. We ran the model five times with different values of C^l. The key results of the optimization are in Table 4.4.

Table 4.4: Effect of late delivery penalty cost

Value of C^l	1	3	5	10	15
Optimality gap (%)	6.08	8.39	7.77	6.58	6.11
Parcels delivered in time (%)	95.1	96.8	97.1	97.5	97.5
Transportation cost	15456.0	20426.0	20062.0	23982.0	25088.0
Sorting cost	63864.75	65130.44	66581.37	67354.39	67951.35
Late delivery cost	10477.38	20263.47	31100.04	53266.69	78011.21
Objective cost	89798.13	105819.9	117743.41	144603.08	171050.56

As seen from the table, the model behaviour seems to respond reasonably to the late delivery penalty cost. The on-time delivery of the parcels is more a question of transportation than a question of sorting. Transport costs increase when more parcels are delivered on time in order to avoid incurring big penalty costs. By looking at the transportation variables y_{tij} and y_{til} we could be able to see in more detail how more parcels were being transported on time. Sorting costs however do not change much while using the different late delivery penalty costs. Late delivery cost increases logically with the value of C^l but not with the same phase since more parcels are delivered in time.

In addition, it seems that the model has a limit on how many parcels can be delivered in time. This might be due to the values of s_{ti} (number of parcels entering to sorting center i at time interval t) obtained from data. Some parcels simply enter the network so late that their timely transportation for example from Vantaa to Oulu becomes impossible.

The analysis of the late delivery cost is also important because the provided data does not give the real value of that cost. When other results were obtained the late delivery cost was set to be 5. It might not be straightforward to decide what value of the late delivery penalty should be used when model is run, but these results give insight and thoughts into how much improving the service level, or the percentage of parcels delivered on time, will cost.

Chapter 5
Discussion

5.1 Reflections on the literature

Our final optimization model may be classified using the taxonomy provided in [1] as a manufacture scheduling and vehicle scheduling and routing problem with time windows. Manufacture scheduling corresponds to the sorting decisions done by the model and vehicle scheduling and routing is in essence the transportation plan provided by the decision variables in the solution. As one in a family of vehicle routing problems using time constraints, our model is also NP hard. The formulation of the model is done using mixed integer linear programming following the generalization provided by Clarke et al. [6] for the vehicle routing problem. It is similar to the pick-up and delivery problem with time windows presented in Lau et al. [15], but due to the nature of the assumptions we use, our model is more simplified. The use of similar solution approaches as Lau was thus considered. However, the size of our model, namely only optimizing for six distribution centers and 33 terminals, was considered to be sufficiently small for using exact methods of mixed integer programming instead.

The choice of constructing the MILP model was done fairly early in the modeling process. However, due to the size of the model and the solver algorithms used by the Gurobi solver we quickly decided to also settle for approximate solutions in order to keep computation times at an acceptable level. This decision corresponds with existing literature as presented in Chapter 2. An improvement to the solutions provided by our model could possibly have been obtained by implementing the proposed algorithms and heuristics found in literature. However, the construction of the model was the main emphasis of this project and thus the choice was made to focus on constructing firstly the model and subsequently implementing a ready commercially available solver instead of constructing or implementing some other heuristic approach from scratch. This was done in order to be able to provide Posti with a tool they could start testing and using on project completion. Thus, using Gurobi was agreed on together with Posti.

Note however that the choice of using Gurobi implicitly contains some of the approaches provided in literature. The standard branch-and-bound approach used by Gurobi starts with an initial solution provided by solving the relaxed linear problem without integrality requirements. This solution is then improved through pruning and cutting planes. Furthermore, several heuristics are implemented within Gurobi to handle situations where the algorithm gets stuck or the MILP problem is too difficult to solve using branch-and-bound [13]. This principle follows the approach adopted in Lau et al. [15] where an initial solution to the problem is constructed and then sequentially improved using line searches. A similar heuristic design is also presented as one option in [21].

This model represents a system that can be seen as a dynamic stock and flow of parcels. These kinds of systems have different leverage points [16] as places to intervene the system and finding ways to understand and use the system better. In analyzing the model in Chapter 4, we focused on two of these leverage points: numbers, when we were analyzing the effects of different costs, and stock and flow structures, when we were comparing the baseline version with the route version. If we could seek leverage from more fundamental leverage points in the future, we should aim to analyze other leverage points as well. These points of leverage can, for example, include changing the system structure or evolving it. In practice this means that the currently fixed network structure of the sorting centers and terminals should be broken and reconstructed. Such reconstruction is indeed one of the bonus tasks given to the project for further consideration by Posti, and it is discussed briefly in Section 5.3.

5.2 Assessment of the results

Because of the NP hardness time limits in the running times are needed. Convergence is slow and obtaining good solutions needs time and computational efficiency. One reason for this might be that the problem has many feasible solutions which are optimal. For instance, often it does not matter which of the specific time intervals $t \in T$ is used to send a truck, the truck will still reach its final destination in time. We had some discussion whether we could implement a "the earlier the better" dynamic in the model but implementing this policy would have required significant changes in the model and was thus not included in the project scope.

This project work also points out that the more computing power can be invested for running the model implementation, the better the obtained results will be given a limited optimization time. For example, the model was run with same parameter values when the transport cost was original (in Table 4.3) and the late delivery penalty cost was $C^l = 5$ (in Table 4.4). The solution presented in Table 4.4 gives a better objective cost and also a smaller gap. This happened due to the hardware differences (recall beginning of Chapter 4).

As a whole, the baseline model produced reasonable results. However, two concerns were raised. First concern was the example of suboptimal decision to transport parcels to a terminal even though capacity of the TU was mostly unused. However, this was rather rare. For the most part, parcels were transported while utilising the capacity of the TU, but certainly these are possible outcomes of the model, although which can be suppressed by increasing the optimization time limit. Second concern was transporting parcels through unusual routes. To respond to this issue, a route model was introduced for the purpose of providing more meaningful results.

This project work also tried a heuristic of its own by limiting the set of possible routes parcels can be transported on. Unfortunately, the results from these tests were rather inconclusive. The benefits of this heuristic might become more apparent if the size of the network was increased enough to make the base model struggle even more. This could be experimented with when addressing the first bonus problem, as discussed in Section 5.3, but we were not able to test it as a part of this project.

The behavior of the base model was verified with a couple of sensitivity analyses. Results obtained regarding the late delivery penalty cost propose that the model behaves rationally, which is a good sign. However, the result regarding the transportation unit cost are more contradictory since the variation of the total transportation cost is quite big compared to variation in transportation unit cost. Maybe better result would be obtained here also if more time and computing power was invested in the sensitivity analysis.

We also focused on the visualizations of the results which turned out to be a significant part of the project if measured in amount of code which was produced. We hope these visualizations serve well the further use of the model inside Posti, making interpretation of the results easier.

5.3 Bonus problems from Posti

In the problem description Posti proposed two bonus tasks on top of the one described in Section 1.3. The first one was the implementation of trailer swaps and optimization of the swap locations. Though we were not able to put much effort on this subject, it is possible that the model could be expanded, quite effortlessly even, to address this problem. One way to do this would be to think of potential locations for trailer swaps in advance and add them into the network as additional "distribution centers" that have zero inflow of parcels ($s_{ti} = 0$ always) and no terminals under them. The only information needed would be the driving time from the new locations to all other new locations and actual distribution centers. They would then simply function as intermediate points for transportations, taking in parcels and placing them in outbound transportation queues. If the optimized solutions would use these locations then they could be considered as potential locations for trailer swaps. The downside of this approach is that adding many new locations to the list of distribution centers quickly increases the number of integer variables in the model, so the locations should be added only a few at a time. Additionally, their locations (that is, the driving times) would need to be decided beforehand as the model is not able to propose good locations by itself.

The second bonus problem was to optimize the division of terminals under the distribution centers (i.e. which distribution center is each terminal linked to). As our model is built over a predetermined network structure addressing this problem is more difficult. Creating a direct optimization algorithm for this kind of a clustering task seems like an entirely new problem. However, it is possible to do some testing with our model by modifying the existing choice of clusters and seeing how it affects the optimization results. But again, the benefit of this is limited as it is merely a way to try out clusterings that have to be created in some other way, possibly manually.

Chapter 6
Conclusions

This project work successfully developed a MILP model for parcel sorting and transportation inside Posti's network in Finland. It addresses the problem formulation and corresponding main tasks given by Posti adequately well. It includes a cost function that optimizes the sorting and transportation costs simultaneously as was requested in the tasking. It uses Gurobi as a solution algorithm for the MILP and it may be tested with the data collected and used in the operations run by Posti. Results derived through varying parameters are reported and compared in a way that enables Posti to fine tune the model based on their requirements and available data. The results were assessed to be realistic and usable.

The model is large and the solutions contain a lot of information. For the project group members it was sometimes difficult to find and interpret the most relevant information provided by each scenario. We hope that the subject matter experts within Posti can therefore use that expertise in order to see and understand the relevance of the results provided by our model more clearly and contextually.

Critically evaluating the project process and outcome however highlights that this project suffered from the "hammer and nail syndrome". Since the existing research literature presented MILP models as viable options for model building, the project team cemented the MILP approach quite quickly. This was probably due to the fact that it was both a familiar method for the team members and the problem formulation was easier to construct using a familiar approach. It is possible, yet difficult to accurately assess, whether our chosen approach limited the quality of our solutions compared to, for example, using heuristics in a more explicit way. Thus, it is possible we might have missed more relevant solution methods and more accurate solutions due to our narrow point of view. Luckily our chosen approach was able to provide us with a decent outcome. Implementing a more specific literature based heuristic and correspondingly updating the model is a possible avenue for further research on the topic of Posti's supply chain optimization.

Although our model was based on specific assumptions and limitations, it can serve as a flexible tool for decision making. Parameter values can be altered to test different kinds of scenarios. The model may also be expanded in several ways, including varying the transportation unit capacities, constraining work shifts of the transportation unit drivers, and varying the types and thus by extension also the sorting times and costs for different types of parcels. Another option is the first bonus problem, discussed in Section 5.3. Although our model does not address the setting of optimizing the underlying network structure, as was also discussed in Section 5.3, using optimization tools for finding the best network structure is a very good candidate for future research on this topic.

References

[1] K. Braekers, K. Ramaekers, and I. Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 12 2015. doi: 10.1016/j.cie.2015.12.007.

[2] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, 39:104–118, 2005. doi: 10.1287/trsc.1030.0056.

[3] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005. doi: 10.1287/trsc.1030.0057. URL `https://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0057`.

[4] S. Bunte and N. Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1(4):299–317, 2009. doi: https://doi.org/10.1007/s12469-010-0018-5.

[5] H.-K. Chen, C.-F. Hsueh, and M.-S. Chang. Production scheduling and vehicle routing with time windows for perishable food products. *Computers Operations Research.*, 36(7):2311–2319, July 2009. doi: 10.1016/j.cor.2008.09.010. URL `https://doi.org/10.1016/j.cor.2008.09.010`.

[6] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964. doi: 10.1287/opre.12.4.568. URL `https://doi.org/10.1287/opre.12.4.568`.

[7] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. doi: 10.1287/opre.2.4.393. URL `https://doi.org/10.1287/opre.2.4.393`.

[8] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959. doi: 10.1287/mnsc.6.1.80. URL `https://doi.org/10.1287/mnsc.6.1.80`.

[9] eMarketer. Global ecommerce forecasts, 2018. URL `https://emarketer.com/forecasts`. Last accessed 6.3.2020.

[10] financeonline.com. ecommerce trends, 2018. URL `https://financesonline.com/ecommerce-trends/`. Last accessed 6.3.2020.

[11] G. Ghiani, G. Laporte, and R. Musmanno. *Introduction to Logistics Systems Management*. John Wiley & Sons, 2013.

[12] A. Gumus, B. Kocaoğlu, and Y. Kocaoğlu. Supply chain optimization studies: A literature review and classification. *Doğuş Üniversitesi Dergisi*, 1:79–98, 01 2018. doi: 10.31671/dogus.2018.16.

[13] Gurobi. Mixed-integer programming (mip) – a primer on the basics, 2020. URL https://www.gurobi.com/resource/mip-basics/. Last accessed 15 May 2020.

[14] B. Kallehauge. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7):2307–2330, 2008.

[15] H. C. Lau and Z. Liang. Pickup and delivery with time windows: Algorithms and test case generation. *International Journal on Artificial Intelligence Tools*, 11(03): 455–472, 2002.

[16] D. H. Meadows. *Thinking in systems: A primer*. Chelsea Green Publishing, 2008.

[17] Posti. Posti to boost its growth by increasing the number of parcel lockers in finland to over 4,000, 2019. URL https://www.posti.com/en/media/media-news/2019/posti-to-boost-its-growth-by-increasing-the-number-of-parcel-lockers/. Last accessed 13 May 2020.

[18] Posti. Posti in brief, 2020. URL https://www.posti.com/en/group-information/posti-in-brief/. Last accessed 13 May 2020.

[19] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, Apr. 1987.

[20] Statista. ecommerce outlook in finland 2018, 2018. URL https://www.statista.com/outlook/243/135/ecommerce/finland. Last accessed 6.3.2020.

[21] S. R. Thangiah, I. H. Osman, R. Vinayagamoorthy, and T. Sun. Algorithms for the vehicle routing problems with time deadlines. *American Journal of Mathematical and Management Sciences*, 13(3-4):323–355, 1993.

[22] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM, 01 2002. doi: 10.1137/1.9780898718515.

[23] C. A. Ullrich. Integrated machine scheduling and vehicle routing with time windows. *European Journal of Operational Research*, 227(1):152–165, 2013.

[24] D. Waters. *Logistics: An Introduction to Supply Chain Management*. Ashford Colour Press Ltd, 2003.

[25] B. Werners and T. Wülfing. Robust optimization of internal transports at a parcel sorting center operated by deutsche post world net. *European Journal of Operational Research*, 201(2):419–426, 2010.

Appendix A
Self assessment

A.1 Overall assessment

The task provided by the client organization Posti was clearly defined but ambitious. As such, the division of tasks to main effort and optional "bonus" tasks guided the planning and ways of working well. The objective was also concisely predefined by Posti from the very beginning of the project. This helped the team focus on the relevant project topics very early on. We assess that the project was successful because we were able to answer the main tasks Posti gave us. Unfortunately, we were only able to scratch the surface of the bonus questions with the time we had available.

Having a clear objective also helped to ensure that the project plan was aligned with project scope and realistic. The scope of the project remained unchanged throughout project completion. This supported adhering to the planned project schedule promptly due to the workload estimates also remaining unchanged. Some changes were inevitable however: one major change was required during the end of the project when we decided to shift our focus from comparing the results with the historical realization data to the construction of visualization tools in order to make the optimization model as easy to use and interpret as possible. This change was relevant and a good learning experience for the team: the importance of how results are communicated clearly to the business stakeholders in working life became an apparent and important lesson.

While the objective of the project and its corresponding task list were defined very well, they were not prioritized by the project manager in the project plan. This prioritisation was done halfway through the project when the topics of the project schedule were also updated, but clearer prioritisation earlier in the project could have decreased some of the workload. Overall the amount of workload was still reasonable even if some workload seemed to get weighted and congested just before mid-project deadlines. This could possibly have been avoided if the project manager had reserved more time for the delivarable preparations or stricter pre-deadlines. Perhaps in the future the teaching staff could also find ways to encourage a more even and continuous way of working during the course?

The biggest effect on workload distribution and project management in general was caused by the COVID19 situation and the corresponding remote working. Remote working combined with other effects of the situation reduced the planned meetings between the team and Posti. This was also one key driver behind the decision to focus on delivering a tool where communicating results in a clear manner was prioritised. The dialogue between the team and Posti was still constant despite the reduced amount of meetings between us.

Risks of the project never truly realized. The risks and mitigation strategies identified in the project plan were sufficient. The biggest risk for the group seemed to be if we could develop and implement a relevant end product at all. Luckily, by keeping things simple enough we were able to avoid this risk successfully. However, the schedule accelerated a little towards the end because there seemed to be a slight delay in receiving the actual data from Posti caused by the COVID situation.

It was good to use Telegram group as a quick chat tool within the project group. With it one could ask questions quickly. However, its role increased when we were forced to work remotely. It also seemed that some relevant things could be lost or missed because the chat was quite active. Maybe it would have made working more easily if official decisions and deadlines were separated from the unofficial speculations.

There was room for improvement in how much the group challenged itself. We could have gone deeper in the literature and attempted to find more varying ways to tackle the tasks given by Posti. Settling on using a MILP model enabled us to focus on delivering a usable end product to Posti, but research and implementation of more recent efficient heuristic algorithms could have potentially provided them with a better solution. This different approach might have also provided us with a greater learning experience but it could equally well have also been a very frustrating, not to mention potentially inconclusive, road to take. This was a decision ultimately made by the project manager and Posti.

In hindsight this is a learning point to the teaching staff as well. The client orginization has an expectation of having "their problem solved" during the course, whereas the university places an emphasis on the academic side of the project. Creating novel approaches is very unlikely for a group of students working in a limited timeframe of a single course. Thus, the students will most likely adapt what they have learned so far instead. Exactly as our team now did. This prioritization between a practical implementation and thorough academic review could be done and guided by the teaching staff in a more clear way.

A.2 Success factors

1. Well defined scope and objective
2. Constant communication between team and client
3. Risk mitigation was successful

A.3 Lessons learned

1. Time constraints preceding deliverable deadlines is to be considered in the project planning and scheduling
2. Communication between teaching staff and project manager should have been more active and interactive

3. The "hammer and nail" phenomenon could have been avoided by maintaining a more open mindset during the modelling process